

Peregrine

AssetCenter

プログラマーズリファレンス

© Copyright 2002 Peregrine Systems, Inc.

All rights reserved.

本書に記載されている情報は、Peregrine Systems, Incorporatedが所有し、Peregrine Systems, Inc.の書面による許可なく使用または開示することはできません。本書の一部または全部を、Peregrine Systems, Inc.の事前の書面による許可なく無断で複製することを禁じます。本書に記載されている商品名は、該当する各社の商標または登録商標です。

Peregrine Systems ®およびAssetCenter ®は、Peregrine Systems, Inc.の商標です。

本書で説明されているソフトウェアは、Peregrine Systems, Inc.とエンドユーザ間で締結されるライセンス契約に基づいて提供されます。契約の条項に従って、ソフトウェアを使用する必要があります。Peregrine Systems, Inc.は、本書の内容については一切の責任を負いかねます。また、本書の内容が予告なく変更されることもあります。本書の最終バージョンの日付を確認するには、Peregrine Systems, Inc.のカスタマサポートまでお問合せください。

デモ用データベースと本書の例に使用されている団体名および個人名は架空のものであり、本ソフトウェアの使用方法を説明するためのものです。現在、過去を問わず、実在する団体や個人とのいかなる類似もまったくの偶然によるものです。

本製品に関する技術情報の請求、またはライセンスをお持ちの製品に関するマニュアル類の請求については、Peregrine Systemsのカスタマサポート ( support@peregrine.com ) までお寄せください。

本マニュアルに関するご意見やご要望は、Peregrine Systems, Inc.の出版部 ( doc\_comments@peregrine.com ) までお寄せください。

本書の内容は、ライセンス契約に基づくプログラムのバージョン4.2.0に適用されます。

AssetCenter

Peregrine Systems, Inc.  
Worldwide Corporate Campus and Executive Briefing Center  
3611 Valley Centre Drive San Diego, CA 92130  
Tel 800.638.5231 or 858.481.5000  
Fax 858.481.1751  
www.peregrine.com



# 目次

I. はじめに . . . . .	17
1. 関数の分類 . . . . .	19
関数の種類 . . . . .	19
関数の用途 . . . . .	20
関数の機能 . . . . .	20
2. 表記法と形式 . . . . .	21
表記法 . . . . .	21
スクリプト内での「日付+時間」型定数のフォーマット . . . . .	22
Duration (時間) 定数の形式 . . . . .	23
3. 定義 . . . . .	25
関数の定義 . . . . .	25
CurrentUser仮想リンクの定義 . . . . .	26
ハンドルの定義 . . . . .	26
エラーコードの定義 . . . . .	27
4. 関数とパラメータのデータ型 . . . . .	29
データ型のリスト . . . . .	29

関数の型 . . . . .	30
パラメータの型 . . . . .	30
<b>II. APIの使用 . . . . .</b>	<b>31</b>
5. はじめに . . . . .	33
注意事項 . . . . .	33
インストール . . . . .	34
6. 操作手順 . . . . .	35
7. 基本概念とプログラムの例 . . . . .	37
基本概念 . . . . .	37
日付と時刻の処理 . . . . .	38
プログラムの例 (1) . . . . .	38
プログラムの例 (2) . . . . .	39
<b>III. 関数の説明 . . . . .</b>	<b>41</b>
8. 関数の説明 . . . . .	43
Abs() . . . . .	43
AmActionDde() . . . . .	44
AmActionExec() . . . . .	46
AmActionMail() . . . . .	47
AmActionPrint() . . . . .	49
AmActionPrintPreview() . . . . .	50
AmActionPrintTo() . . . . .	50
AmAddAllPOLinesToInv() . . . . .	51
AmAddCatRefAndCompositionToPOOrder() . . . . .	52
AmAddCatRefToPOOrder() . . . . .	54
AmAddEstimLinesToPO() . . . . .	55
AmAddEstimLineToPO() . . . . .	56
AmAddPOLineToInv() . . . . .	57
AmAddPOOrderLineToReceipt() . . . . .	58
AmAddReceiptLineToInvoice() . . . . .	59
AmAddReqLinesToEstim() . . . . .	61
AmAddReqLinesToPO() . . . . .	62
AmAddReqLineToEstim() . . . . .	63
AmAddReqLineToPO() . . . . .	64
AmAddRequestLineToPOOrder() . . . . .	65
AmAddTemplateToPOOrder() . . . . .	66
AmAddTemplateToRequest() . . . . .	67

AmBusinessSecondsInDay()	69
AmCalcConsolidatedFeature()	70
AmCalcDepr()	71
AmCbkReplayEvent()	72
AmCheckTraceDone()	73
AmCleanup()	74
AmClearLastError()	75
AmCloseAllChildren()	76
AmCloseConnection()	76
AmCommit()	77
AmComputeAllLicAndInstallCounts()	78
AmComputeLicAndInstallCounts()	79
AmConnectTrace()	80
AmConvertCurrency()	81
AmConvertDateBasicToUnix()	83
AmConvertDateIntlToUnix()	84
AmConvertDateStringToUnix()	85
AmConvertDateUnixToBasic()	86
AmConvertDateUnixToIntl()	87
AmConvertDateUnixToString()	88
AmConvertDoubleToString()	90
AmConvertMonetaryToString()	91
AmConvertStringToDouble()	92
AmConvertStringToMonetary()	93
AmCounter()	94
AmCreateAssetPort()	95
AmCreateAssetsAwaitingDelivery()	96
AmCreateCable()	97
AmCreateCableBundle()	99
AmCreateCableLink()	100
AmCreateDelivFromPO()	102
AmCreateDevice()	103
AmCreateDeviceLink()	104
AmCreateEstimFromReq()	106
AmCreateEstimsFromAllReqLines()	107
AmCreateInvFromPO()	108
AmCreateLink()	109
AmCreatePOFromEstim()	110
AmCreatePOFromReq()	111
AmCreatePOOrderFromRequest()	112
AmCreatePOOrdersFromRequest()	113
AmCreatePOsFromAllReqLines()	114
AmCreateProjectCable()	115
AmCreateProjectDevice()	116

AmCreateProjectTrace()	117
AmCreateReceiptFromPOrder()	119
AmCreateRecord()	120
AmCreateRequestToInvoice()	121
AmCreateRequestToPOrder()	122
AmCreateRequestToReceipt()	124
AmCreateReturnFromReceipt()	125
AmCreateTraceHist()	126
AmCryptPassword()	127
AmCurrentDate()	128
AmCurrentIsoLang()	129
AmCurrentLanguage()	130
AmCurrentServerDate()	131
AmDateAdd()	132
AmDateAddLogical()	134
AmDateDiff()	135
AmDbGetDate()	136
AmDbGetDouble()	137
AmDbGetList()	138
AmDbGetListEx()	140
AmDbGetLong()	141
AmDbGetPk()	142
AmDbGetString()	143
AmDbGetStringEx()	145
AmDeadLine()	147
AmDecrementLogLevel()	148
AmDefAssignee()	149
AmDefaultCurrency()	150
AmDefEscalationScheme()	151
AmDefGroup()	152
AmDeleteLink()	154
AmDeleteRecord()	155
AmDisconnectTrace()	156
AmDuplicateRecord()	157
AmEndOfNthBusinessDay()	158
AmEnumValList()	159
AmEvalScript()	161
AmExecTransition()	162
AmExecuteActionById()	163
AmExecuteActionByName()	164
AmExportDocument()	165
AmFindCable()	166
AmFindDevice()	168
AmFindRootLink()	169

AmFindTermDevice()	170
AmFindTermField()	171
AmGenSqlName()	172
AmGetCatRef()	174
AmGetCatRefFromCatProduct()	175
AmGetComputeString()	176
AmGetCurrentNTDomain()	177
AmGetCurrentNTUser()	178
AmGetFeat()	179
AmGetFeatCount()	180
AmGetField()	181
AmGetFieldCount()	182
AmGetFieldDateValue()	183
AmGetFieldDescription()	184
AmGetFieldDoubleValue()	185
AmGetFieldFormat()	186
AmGetFieldFormatFromName()	188
AmGetFieldFromName()	189
AmGetFieldLabel()	190
AmGetFieldLabelFromName()	191
AmGetFieldLongValue()	192
AmGetFieldName()	193
AmGetFieldRights()	194
AmGetFieldSize()	195
AmGetFieldSqlName()	196
AmGetFieldStrValue()	197
AmGetFieldType()	199
AmGetFieldUserType()	201
AmGetForeignKey()	203
AmGetIndex()	204
AmGetIndexCount()	205
AmGetIndexField()	206
AmGetIndexFieldCount()	207
AmGetIndexFlags()	208
AmGetIndexName()	209
AmGetLink()	210
AmGetLinkCardinality()	211
AmGetLinkCount()	212
AmGetLinkDstField()	213
AmGetLinkFeatureValue()	214
AmGetLinkFromName()	215
AmGetLinkType()	216
AmGetMainField()	217
AmGetMemoField()	218

AmGetNextAssetPin()	219
AmGetNextAssetPort()	220
AmGetNextCableBundle()	221
AmGetNextCablePair()	223
AmGetNTDomains()	224
AmGetNTMachinesInDomain()	225
AmGetNTUsersInDomain()	226
AmGetPOLinePrice()	227
AmGetPOLinePriceCur()	228
AmGetPOLineReference()	229
AmGetRecordFromMainId()	230
AmGetRecordHandle()	231
AmGetRecordId()	232
AmGetRelDstField()	233
AmGetRelSrcField()	234
AmGetRelTable()	234
AmGetReverseLink()	235
AmGetSelfFromMainId()	236
AmGetSourceTable()	237
AmGetTable()	238
AmGetTableCount()	239
AmGetTableDescription()	240
AmGetTableFromName()	241
AmGetTableLabel()	242
AmGetTableName()	243
AmGetTableRights()	244
AmGetTableSqlName()	245
AmGetTargetTable()	246
AmGetTrace()	247
AmGetTraceFromHist()	249
AmGetTypedLinkField()	250
AmGetVersion()	251
AmHasAdminPrivilege()	252
AmHasRelTable()	253
AmImportDocument()	254
AmIncrementLogLevel()	255
AmInsertRecord()	256
AmInstantiateReqLine()	257
AmInstantiateRequest()	258
AmlsConnected()	259
AmlsFieldForeignKey()	260
AmlsFieldIndexed()	261
AmlsFieldPrimaryKey()	262
AmlsLink()	263

AmIsTypedLink()	264
AmLastError()	265
AmLastErrorMsg()	266
AmListToString()	267
AmLog()	268
AmLoginId()	269
AmLoginName()	270
AmMapSubReqLineAgent()	271
AmMoveCable()	272
AmMoveDevice()	273
AmMsgBox()	275
AmOpenConnection()	276
AmOpenScreen()	276
AmPagePath()	278
AmProgress()	278
AmQueryCreate()	279
AmQueryExec()	280
AmQueryGet()	281
AmQueryNext()	282
AmQuerySetAddMainField()	283
AmQuerySetFullMemo()	284
AmQueryStartTable()	285
AmQueryStop()	286
AmReceiveAllPOLines()	287
AmReceivePOLine()	288
AmRefreshAllCaches()	289
AmRefreshLabel()	290
AmRefreshProperty()	291
AmRefreshTraceHist()	292
AmReleaseHandle()	293
AmRemoveCable()	294
AmRemoveDevice()	296
AmReturnAsset()	297
AmReturnContract()	298
AmReturnPortfolioItem()	299
AmReturnTraining()	300
AmReturnWorkOrder()	301
AmRevCryptPassword()	303
AmRgbColor()	304
AmRollback()	305
AmSetFieldDateValue()	306
AmSetFieldDoubleValue()	307
AmSetFieldLongValue()	308
AmSetFieldStrValue()	309

AmSetLinkFeatureValue()	310
AmSetProperty()	311
AmShowCableCrossConnect()	312
AmShowDeviceCrossConnect()	313
AmSqlTextConst()	314
AmStartTransaction()	315
AmStartup()	316
AmTableDesc()	316
AmTaxRate()	318
AmUpdateDetail()	319
AmUpdateRecord()	320
AmValueOf()	321
AmWizChain()	322
AmWorkTimeSpanBetween()	323
AppendOperand()	324
ApplyNewVals()	326
Asc()	327
Atn()	328
BasicToLocalDate()	329
BasicToLocalTime()	330
BasicToLocalTimeStamp()	331
Beep()	332
CDBI()	333
ChDir()	334
ChDrive()	335
Chr()	336
CInt()	337
CLng()	338
Cos()	339
CountOccurrences()	340
CountValues()	342
CSng()	343
CStr()	344
CurDir()	345
CVar()	346
DaContext()	347
DaCopy()	349
DaDbDeleteList()	350
DaDbGetList()	352
DaDbSetList()	353
DaDelete()	355
DaDownload()	357
DaDumpContext()	358
DaExec()	360

DaExecAction()	361
DaExecuteActionByName()	363
DaFileATime()	364
DaFileCRC()	365
DaFileCTime()	366
DaFileLanguage()	367
DaFileMTime()	368
DaFileSize()	369
DaFileType()	370
DaFileVersion()	371
DaFind()	372
DaFindNext()	373
DaFirstEnv()	375
DaGetEnv()	376
DaGetFileInfo()	378
DaImpersonate()	379
DaMkDir()	381
DaMove()	382
DaNetIpFromName()	384
DaNetNBName()	385
DaNetPing()	386
DaNetWakeOnLan()	387
DaNetWinAddressByName()	388
DaNextEnv()	389
DaNTFileCopyTo()	391
DaNTFileCreateDir()	392
DaNTFileDelete()	393
DaNTFileDeleteDir()	393
DaNTFileDirCopyTo()	394
DaNTFileDirDownload()	395
DaNTFileDirUpload()	396
DaNTFileDownload()	397
DaNTFileUpload()	398
DaNTRegistryLMAddStringValue()	399
DaNTRegistryLMCreateKey()	400
DaNTRegistryLMDeleteKey()	400
DaNTRegistryLMDeleteValue()	401
DaNTRegistryLMGetLongValue()	402
DaNTRegistryLMGetStringValue()	403
DaNTRegistryLMSetLongValue()	404
DaNTRegistryLMSetStringValue()	405
DaNTServiceInstall()	406
DaNTServiceStart()	407
DaNTServiceStatus()	408

DaNTServiceStop()	409
DaNTServiceUninstall()	410
DaNTWMIExecMethod()	411
DaNTWMIExecQuery()	412
DaNTWMIGetCurrentArrayValue()	414
DaNTWMIGetCurrentPropertyValue()	415
DaNTWMIGetInstanceCount()	416
DaNTWMIGetPropertyValue()	417
DaNTWMIGetTotalPropertiesValue()	419
DaNTWMINextItem()	421
DaNTWMIResetEnumeration()	421
DaRegCreateKey()	422
DaRegDeleteKey()	423
DaRegExec()	424
DaRegGetValue()	425
DaRegOutputValue()	426
DaRegSetValue()	427
DaRegStrValue()	428
DaRegVarValue()	429
DaRename()	430
DaReturnValue()	431
DaRmdir()	432
DaSendMail()	434
DaSetContext()	435
DaSetOption()	437
DaSetReturnValue()	439
Date()	440
DateSerial()	441
DateValue()	442
DaTrackingDelete()	444
DaTrackingGet()	445
DaTrackingSet()	446
DaTrackingTest()	447
DaUpload()	448
DaWait()	450
Day()	451
EscapeSeparators()	452
ExeDir()	453
Exp()	454
ExtractValue()	455
FileCopy()	457
FileDateTime()	458
FileExists()	458
FileLen()	459

Fix()	460
FormatDate()	461
FormatResString()	463
FV()	464
GetListItem()	465
Hex()	467
Hour()	468
InStr()	469
Int()	470
IPMT()	471
IsNumeric()	473
Kill()	473
LCase()	474
Left()	476
LeftPart()	477
LeftPartFromRight()	478
Len()	480
LocalToDate()	481
LocalToBasicTime()	482
LocalToBasicTimeStamp()	483
LocalToUTCDate()	484
Log()	485
LTrim()	486
MakeInvertBool()	487
Mid()	488
Minute()	489
MkDir()	491
Month()	491
Name()	493
Now()	494
NPER()	494
Oct()	496
ParseDate()	497
ParseDMYDate()	498
ParseMDYDate()	499
ParseYMDDate()	500
PMT()	501
PPMT()	503
PV()	504
Randomize()	506
RATE()	507
RemoveRows()	509
Replace()	510
Right()	512

RightPart()	513
RightPartFromLeft()	514
Rmdir()	516
Rnd()	517
RTrim()	518
Second()	520
SetSubList()	521
Sgn()	522
Shell()	523
Sin()	524
Space()	525
Sqr()	527
Str()	528
StrComp()	529
String()	530
SubList()	531
Tan()	533
Time()	534
Timer()	535
TimeSerial()	536
TimeValue()	537
ToSmart()	538
Trim()	539
UCase()	540
UnEscapeSeparators()	542
Union()	543
UTCToLocalDate()	544
Val()	545
WeekDay()	546
Year()	547
<b>IV. 索引</b>	<b>549</b>
<b>使用可能な関数 - 機能：全種</b>	<b>551</b>
<b>使用可能な関数 - 機能：調達</b>	<b>559</b>
<b>使用可能な関数 - 機能：ケーブル</b>	<b>561</b>
<b>使用可能な関数 - 機能：アクション</b>	<b>563</b>
<b>使用可能な関数 - 機能：組み込み</b>	<b>565</b>

使用可能な関数 - 機能：ウィザード . . . . .	567
使用可能な関数 - 機能：認証 . . . . .	569
使用可能な関数 - 機能：リモート実行 . . . . .	571
使用可能な関数 - 機能：ファイル管理 . . . . .	573
使用可能な関数 - 機能：コンテキスト . . . . .	575
使用可能な関数 - 機能：レジストリ . . . . .	577
使用可能な関数 - 機能：データベース . . . . .	579
使用可能な関数 - 機能：トラッキング . . . . .	581
使用可能な関数 - 機能：WMI . . . . .	583
使用可能な関数 - 機能：ネットワーク . . . . .	585





パート | はじめに

---



# 1 | 関数の分類

---

関数は、次の3通りの方法で分類することができます。

- 関数の種類
- 関数の用途
- 関数の機能

## 関数の種類

AssetCenter環境で使用できる関数は、複数のグループに分類されます。

- AssetCenterで認識される関数：主にソフトウェアのスク립ト（BASIC）を記述できる部分で使います。
- AssetCenter APIで認識される関数：外部のツールから呼び出される関数、または高度言語で書くプログラムで使う関数です。

これらの関数は、それぞれ完全に独立しているわけではありません。例えば、AssetCenter API関数の中には、ソフトウェアのBASICスク립トで使えるものがあります。このようにBASICスク립トでも使えるAssetCenter API関数のことを「公開されている」関数といいます。このような関数のシンタックスが変わることはありますが、関数で実行される処理は同じです。

## 関数の用途

本マニュアルで説明されている関数は、少なくとも次の場所の1つで使用可能です。

- AssetCenter APIライブラリ。これらの関数は、特にGet-Itアプリケーションの開発で使用可能です。
- フィールドやリンクの設定スクリプト（ポップアップメニューで【**オブジェクトの設定**】またはAssetCenter Database Administratorを選択します）、または特殊フィールドの【**計算スクリプト**】（SQL名：memScript）内。
  - デフォルト値
  - 必須属性
  - 履歴の保持
  - 読み取り専用属性
  - ...
- 「スクリプト」タイプのアクション
  - スクリプトアクションの【**アクションのスクリプト**】（SQL名：Script）フィールドで定義されるスクリプト
- AssetCenterウィザード
  - ウィザードの「FINISH.DO」スクリプト
  - ノードのプロパティの値を定義するスクリプト

## 関数の機能

各関数は1つまたは複数の機能ドメインに関連付けられています。機能ドメインは、関数が実行する演算の属性を指します。以下の機能ドメインがあります。

- 組み込み：標準のBasic関数や変換関数、文字列の操作など
- テクニカル：データベースへの接続、テーブル、フィールド、リンク、インデックス、レコード、クエリの管理
- 業務：専門分野に関する一般的な関数
- ケーブル
- 調達
- 経費付替え
- ウィザード
- アクション
- グラフィックス

## 2 | 表記法と形式

本章では、次の事項について説明します。

- 表記法
- スクリプト内での「日付+時間」型定数のフォーマット
- Duration（時間）定数の形式

### 表記法

関数のシンタックスと用例の表記法は以下の表の通りです。

[ ]	大括弧は、オプションのパラメータを示します。実際にコマンドのパラメータを入力するときは、大括弧は必要ありません。 ただし、BASICスクリプトでは、次の例のように、データへのパスを大括弧で囲みます。 [Link.Link.Field]
<>	山形括弧は、パラメータの短い説明（英語の略語）を示します。実際にパラメータを入力する時は、山形括弧を使わずに、括弧内にあるテキストに該当する情報のみを入力してください。
{ }	中括弧は、ノードの定義または1プロパティ用の複数言語のスクリプトブロックの定義を囲みます。

縦線（パイプ文字）は、中括弧に囲まれた複数のパラメータを区切る場合に使います。

特別な意味のあるテキストの表記法は以下の通りです。

固定幅の文字	DOSコマンド、関数のパラメータ、および日付形式
例	コードやコマンドの例
...	コードまたはコマンドの省略
<b>[オブジェクト名]</b>	フィールド、タブ、メニュー、キー名など
注意:	重要な注意事項
注意	

## スクリプト内での「日付+時間」型定数のフォーマット

ユーザが定義している表示形式に関係なく、スクリプトで日付を記述するときは、次のような国際標準形式を使います。

yyyy/mm/dd hh:mm:ss

例

```
RetVal="1998/07/12 13:05:00"
```

 注意:

年月日の区切りに、ハイフン (-) を使うこともできます。

## 日付について

内部BASICでの日付と外部ツールからの日付は形式が異なります。

- BASICでは、国際標準形式、またはDouble（倍精度）型の浮動小数点数で記述します。後者の場合、整数部分は1899年12月30日の午前0時から数えて現在まで経過した日数を表し、小数部分は本日の午前0時から現在まで経過した時間の1日（86400秒）に対する割合（現在までの経過秒数を86400で割ったもの）を表します。
- UNIXでは、Long（倍長）型の整数で記述します。現地時間に関係なく、UTC（協定世界時）で1870年1月1日の午前0時から現在までに経過した秒数を示します。

## Duration (時間) 定数の形式

スクリプトでは、時間を秒単位で記述して保存します。例えば、Duration型フィールドのデフォルト値を3日に設定するには、次のスクリプトを使います。

```
RetVal=259200
```

同様に、AmWorkTimeSpanBetween()などの時間を計算する関数も、秒単位の時間を返します。

---

### 注意:

AssetCenterの会計計算では、最も一般的な計算方法を使います。この方法では、1年は12ヶ月、1月は30日として計算するので、1年は360日になります。

---



# 3 | 定義

---

本章では、主要な用語の定義について説明します。

以下の用語の定義を説明します。

- 関数の定義
- CurrentUser仮想リンクの定義
- ハンドルの定義
- エラーコードの定義

## 関数の定義

「関数」とは、なんらかの処理を実行し、値をユーザに返すプログラムです。返される値を「戻り値」または「戻りコード」といいます。

AssetCenter内部Basicで関数を呼び出すシンタックスの例は次の通りです。

```
AmConvertCurrency(strSrcName As String, strDstName As String, dVal As Double) As Double
```

上記と同じ関数をAssetCenter APIから呼び出すシンタックスは次の通りです。

```
double AmConvertCurrency(long hApiCnxBase,long ltm, const char *pszSrcName, const char *pszDstName,double dVal)
```

## CurrentUser仮想リンクの定義

### 定義

CurrentUserは、すべてのテーブルから出発し、現在のユーザに対応する部署と従業員テーブル内のレコードに到達するリンクと考えられます。

- 「CurrentUser」という形式を使うと、仮想リンクは現在のユーザのレコードを検索し、ユーザのID番号を返します。
- 「CurrentUser.Field」という形式を使うと、現在のユーザのフィールドの値を返します。

#### 注意:

仮想リンクは、フィールドとリンクのリストには表示されません。そのため、CurrentUserは、AssetCenter内のスクリプトの表示 / 編集用ウィンドウには直接表示されません。手動で「CurrentUser」と入力する必要があります。

### 等価関数

AmLoginName()とAmLoginId()関数は、それぞれ現在のユーザの「名前 (SQL名: Name)」と「ID (SQL名: IPersId)」を返し、CurrentUserから派生した関数と見なされます。これらの関数には、次のような関係が成り立ちます。

- AmLoginName()=[CurrentUser.Name]
- AmLoginId()=[CurrentUser.IEmpIDeptId]

## ハンドルの定義

「ハンドル」とは、オブジェクト固有の識別子です。AssetCenterでは、フィールド、リンク、インデックス、クエリ、レコード、テーブル、または接続のことをオブジェクトと呼びます。ハンドルは、32ビット整数型 (Long型) の値を取ります。

#### 注意:

NULLは、有効なハンドルの値ではありません。  
外部ツールからも (データベースの) 接続ハンドルにアクセスできます。

## エラーコードの定義

関数を正しく実行できなかった場合は、エラーコードが返ります。

### 外部ツールの場合

外部ツールでは、AmLastError()関数とAmLastErrorMsg()関数を使って、それぞれエラーコードとそのコードに付いているエラーメッセージを取得することができます。エラーをクリアするには、AmClearLastError()関数を使います。

 **注意:**

新しい関数を呼び出すと、常にその前に発生したエラーとエラーメッセージがクリアされます。

### 内部

内部 (Basicスクリプト内など) では、最後のエラーコードとエラーの説明を、Err.Number関数とErr.Description関数を使って取得します。

 **注意:**

内部的にはエラーの管理をプログラムする必要はありません。問題のあるスクリプトは停止し、必要に応じてデータベースのロールバックが実行されます。

Err.Raise関数を使用すると、エラーメッセージを故意に発生させることができます。シンタックスは以下の通りです。

```
Err.Raise (<エラー番号>, <エラーメッセージ>)
```

 **注意:**

レコードの作成や変更が、テーブルの「有効性」に関するフィールド値により無効になる場合は、Err.Raise関数でエラーメッセージを発生させ、ユーザに知らせる方が賢明です (コード12006または12007)。エラーメッセージがないと、ユーザはレコードの変更や作成が不可能な理由を理解できません。

頻繁に使用されるエラーコードは以下の表の通りです。

#### エラーコード

12001

#### 説明

未定義のエラー

<b>エラーコード</b>	<b>説明</b>
12002	関数用の間違ったパラメータ
12003	無効なハンドルまたは削除されるオブジェクト
12004	使用可能なデータがない場合。このエラーは特にクエリの実行時に発生します。クエリの結果がデータを返さない場合に、このエラーが発生します。
12005	データベースサーバの内部エラー
12006	無効な値（パラメータに不適切なタイプ、など）
12007	無効なレコード（必須フィールドに値が入力されていない、など）
12008	データベースへのアクセス権限上の問題
12009	古い関数または実施されていない関数
12010	データベースへの接続最大数を越えた場合

# 4 | 関数とパラメータのデータ型

本章では、次の事項について説明します。

- データ型のリスト
- 関数の型
- パラメータの型

## データ型のリスト

関数とパラメータで使用可能なデータ型とその説明は、以下の表の通りです。

データ型	説明
Integer (整数)	-32,768 ~ +32,767の整数
Long (倍長整数)	-2,147,483,647 ~ +2,147,483,646の整数
Double (倍精度)	8バイト浮動小数点数
String (文字列)	任意の文字からなるテキスト
Date (日付)	日付または日付+時刻
Variant (可変)	汎用の任意の型

 **注意:**

これらのデータ型の一部は外部ツールからは使用不可能です。Long、DoubleとString型のみが使用可能です。Variant型は存在しません。また、IntegerとDate型のオブジェクトはLong型で表現されます。

## 関数の型

関数の型は、その関数の返す値の型に対応しています。プログラムのコンパイルエラーやランタイムエラーが発生するのを防ぐために、正しい型の関数を適切な場所で使う必要があります。

例えば、フィールドのデフォルト値を定義する時に、そのフィールドのデータ型と異なる型の値を返す関数を使うことはできません。例えば、下の関数を使ってDate（日付）型やDate+Time（日付+時刻）型のフィールドのデフォルト値を定義すると、エラーが発生します。

```
RetVal=AmLoginName()
```

これは、AmLoginName()関数は、接続されているユーザの名前を文字列（String型）で返すためです。文字列型の戻り値は、Date（日付）型やDate+Time（日付+時刻）型のフィールドでは使えないため、エラーメッセージが表示されます。

## パラメータの型

関数で使うパラメータには型が決まっており、関数を正しく実行するためには、正しい型のパラメータを使用する必要があります。関数のシンタックス内では、パラメータの型を示す接頭辞がパラメータの先頭に付いています。このリファレンスに記載されている接頭辞は、関数のシンタックス（APIまたはBASIC）によって異なることに注意してください。それぞれAPIシンタックスとBASICシンタックスで使われている接頭辞は、次の表の通りです。

型	API関数のシンタックスで使われている接頭辞	BASIC関数のシンタックスで使われている接頭辞
Integer（整数）	i	i
Long（倍長整数）	h（ハンドル）l（数値）	l
Double（倍精度）	d	d
String（文字列）	char *psz	str
Date（日付）	ltime	dt
Variant（可変）	v	v



# II | APIの使用

---

パート



# 5 | はじめに

---

AssetCenter APIは、Windows 95/ 98、Windows NTまたはWindows 2000上で使える32ビットDLLとして用意されています。

AssetCenter APIは、次の環境で使えることが確認されています。

- Visual Basic 4.0、5.0、および6.0
- Visual C++ 4.0、5.0、および6.0
- VBA ( Visual Basic for Applications ) を使用するMicrosoft全製品

---

 **注意:**

APIは、外部DLLの使用を許可するすべてのツールと互換性があります。

---

## 注意事項

AssetCenter APIを使う前に、AssetCenterの概念と用語を理解しておくことをお勧めします。特に、データベースの構造に関する知識が必要です。

データベースの構造の詳細は、AssetCenterのインストール先フォルダの「doc/Infos」サブフォルダにあるマニュアル『管理』の、「データベースの標準記述ファイル」の章、および「Database.txt」と「Tables.txt」ファイルを参照してください。

## インストール

AssetCenter APIを使用する前に、AssetCenterの完全インストールを実行してください。これにより、使用しているコンピュータからデータベースにアクセスできるかどうかを簡単に確認できるようになり、またデータベースの作成や設定が可能になります。APIは、AssetCenterと同じデータベース層と設定情報を使って、データソースにアクセスします。このため、APIで問題が発生した場合は、AssetCenterのGUIで問題の原因を探ることができます。

AssetCenterでの開発環境を設定するための一般的な手順は、次の通りです。

- AssetCenterの32ビットバージョンとAssetCenter APIパッケージをインストールします。
- AssetCenterを使って、データソースを設定し、データベースを開きます。
- 開発環境で、AssetCenter API関数を呼び出します。

デモ用データベースか、エラーが発生してデータが壊れても差し支えないデータソースを使って、AssetCenter APIを試用してください。

## 6 | 操作手順

---

一般的な処理手順は以下の通りです。

1. AQL文を使ってクエリを作成します。

```
SELECT AssetTag, User.Name, Supervisor.Name FROM  
amPortfolio
```

---

 注意:

AssetCenter Exportを使ってAQLクエリを作成することもできます。

---

2. クエリの結果を検索し、特定の項目の必要なハンドルを取得します。
3. 取得したハンドルを使って、レコード情報を更新します。
4. トランザクション全体をコミット（完了）するか、ロールバック（キャンセル）します。



# 7 | 基本概念とプログラムの例

本章では、次の事項について説明します。

- 基本概念
- 日付と時刻の処理
- プログラムの例 (1)
- プログラムの例 (2)

## 基本概念

AssetCenterは、オブジェクト指向で設計されており、このためAPIもオブジェクト指向型の構造です。Windows DLLには、単層構造の「C言語型」APIを使用するという制約点がありますが、AssetCenter APIは、この問題を回避するためにハンドル (32ビット整数) を使ってユーザが作成した各オブジェクトを識別します。このため、オブジェクト指向言語以外のプログラムからでも、AssetCenter オブジェクトモデルにアクセスできるようになっています。

使用するプログラムは、AssetCenter DLLを初期化するために、最初にAmStartUp() 関数を呼び出さなければなりません。また、最後にAmCleanUp()関数を呼び出して終了する必要があります。

AssetCenterのオブジェクトにアクセスする前に、ユーザとAssetCenterデータベースの接続を確立する必要があります。この接続は、「接続」オブジェクトの「ハ

ンドル」で識別されます。このハンドルは、その後データベースと相互作用するすべてのAPI関数で使うことができます。（ハンドルは「hApiCnxBase」パラメータで指定します。）「接続」オブジェクトは、クエリを作成し、レコードにアクセスするのに使用されます。

#### 注意:

すべてのデータベースオブジェクトが接続オブジェクトにリンクされるので、ユーザのアクセス権限情報をチェックできます。

接続が確立したら、まず有効なデータソース名とログイン名/パスワードを使って、接続を開きます。

#### 警告:

AssetCenter APIを介してAssetCenterデータベースに接続するときは、スロットを使います。

## 日付と時刻の処理

Date（日付）型とDate+Time（日付+時刻）型のフィールドの値を読み取るには、次のいずれかの関数を使います。

- AmGetFieldLongValue()：UNIXで使うLong（倍長整数）型のUTC（協定世界時）を返します。
- AmGetFieldStrValue()：Windowsのコントロールパネルで設定されている形式と同じ文字列を返します。Windowsで指定しているタイムゾーンに合わせた日付と時刻が返ります。戻り値を表示する必要がある場合には、この関数を使うことをお勧めします。

## プログラムの例（1）

デモ用データベースへの接続を作成するC言語プログラムの例は、次の通りです。

```
long lCnx ;
lCnx = AmOpenConnection(ACDemo351FRA, Admin , ) ;
```

この例のlCnxは、新しく作成する接続のハンドルです。

次に、作成した接続を使ってクエリを作成したり、データベースにアクセスしたりできます。特定の資産を検索するC言語プログラムの例は、以下の通りです。

```

#include apiproto.h
#define SZ_MODEL_LEN 200
long ICnx ;
long IQuery ;
long IStatus ; /* to store error code */
char szModel[SZ_MODEL_LEN] ;
/* dll initialization */
AmStartup();
/* Open a connection */
ICnx = AmOpenConnection("ACDemo300Eng","Admin", "");
if( ICnx != 0 )
{
    /* Creation of a query object */
    IQuery = AmQueryCreate( ICnx )
    if( IQuery != 0 )
    {
        /* Construction of the result set : all assets from Compaq*/
        IStatus = AmQueryExec(IQuery, "select AssetTag where brand = 'Compaq'")
        /* Navigates through the result set */
        while( !IStatus )
        {
            /* Read the first field (AssetTag) of the current item in the query */
            IStatus = AmGetFieldStrValue(IQuery,0,szModel,SZ_MODEL_LEN-1);
            if( IStatus == 0 )
            {
                printf(' Compaq AssetTag=%s \n',szModel);
                IStatus = AmQueryNext(IQuery);
            }
        }
        /* clean things up */
        AmReleaseHandle(IQuery);
    }
    AmCloseConnection(ICnx);
}
AmCleanup();

```

## プログラムの例 (2)

クエリを使って、データベース内でオブジェクトを検索します。フィールドを更新するときに、クエリからターゲットレコードのハンドルが返されます。次に、特定のAssetCenter APIを使ってレコードを処理します。

特定のレコードハンドルを取得して、フィールドを変更するプログラムの例は、以下の通りです。

```
/* Handles for objects */
long ICnx ;
long IQuery ;
long IStatus ;
long IRecord ;
AmStartup();
ICnx = AmOpenConnection("ACDemo300Eng","Admin", "");
/* Creation of a query object attached to ICnx */
IQuery = AmQueryCreate(ICnx);
/* Mark the starting point of the current transaction */
AmStartTransaction(ICnx);
/* Use a query that matches a single object */
IStatus = AmQueryGet(IQuery, "select model, AssetId where brand = 'Compaq' and bar
code='34234'");
/* Get a record handle to the matching object */
IRecord = AmGetRecordHandle(IQuery);
/* Change the field Field1 with new value spam */
IStatus = AmSetFieldStrValue(IRecord, "Field1", "Spam");
/* Update the change for the current session */
IStatus = AmUpdateRecord(IRecord);
/* Commit all modifications to the database */
IStatus = AmCommit(ICnx);
/* you can release here query and record objects */
/* but closing connection will do it */
/* Close the connection to the database */
AmCloseConnection(ICnx);
AmCleanup();
```

この例では、クエリを使ってレコード特有の識別子（ハンドル）を取得する方法を示しています。ここで使用したクエリでは、オブジェクトを1つだけ検索しますが、AmQueryExec()を使うと、複数のレコードを検索して1つまたは複数のレコードハンドルが返るようにすることもできます。

 **注意:**

この例を簡単にするため、発生する可能性のあるエラーコードは含まれていません。



# III 関数の説明

---

パート



# 8 | 関数の説明

## Abs()

数値の絶対値を返します。

## 内部BASICシンタックス

Function Abs(dValue As Double) As Double

## 用途

バージョン : 3.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- dValue : 絶対値を取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim iSeed as Integer
iSeed = Int((10*Rnd)-5)
RetVal = Abs(iSeed)
```

## AmActionDde()

DDEサーバアプリケーションにDDE要求を送ります。この関数を使うと、DDEリンクにより、AssetCenterから他のアプリケーションをコントロールできます。この関数はDDEタイプのアクションに相当します。

## APIシンタックス

```
long AmActionDde(char *strService, char *strTopic, char *strCommand, char
*strFileName, char *strDirectory, char *strParameters, char *strTable, long IRecordId);
```

## 内部BASICシンタックス

```
Function AmActionDde(strService As String, strTopic As String, strCommand As String,
strFileName As String, strDirectory As String, strParameters As String, strTable As String,
IRecordId As Long) As Long
```

## 用途

バージョン：3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strService：呼び出す実行可能ファイルのDDEサービスの名前。実行可能ファイルのDDEサービスの一覧については、その実行可能ファイルのマニュアルを参照してください。
- strTopic：実行するDDEアクションのトピック（コンテキスト）
- strCommand：実行する外部アプリケーションのコマンド。外部アプリケーションで定義されているシンタックスに従って指定する必要があります。
- strFileName：サービスがメモリにない場合は、サービスを起動する実行可能ファイルの名前（または、Windowsのファイルマネージャで実行可能ファイルとして設定されているファイルの名前）をこのパラメータで指定して、サービスを起動する必要があります。
- strDirectory：strFileNameで指定したファイルのパス
- strParameters：サービスの起動時にサービスをアクティブにする実行可能ファイルに渡す様々なパラメータ
- strTable：アクションのコンテキストを指定する時に追加するパラメータ。アクションを適用するレコードが入っているテーブルのSQL名を示します。
- IRecordId：アクションのコンテキストを指定するときに追加するパラメータ。アクションを適用するレコードが入っているテーブルのSQL名を指定します。

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmActionExec()

「.exe」、「.com」、「.bat」、「.pif」のいずれかのアプリケーションを起動します。また、Windowsのファイルマネージャで実行可能ファイルとして設定している拡張子の文書を参照できます。この関数は、「実行可能ファイル」タイプのアクションに相当します。

## APIシンタックス

```
long AmActionExec(char *strFileName, char *strDirectory, char *strParameters, char *strTable, long IRecordId);
```

## 内部BASICシンタックス

```
Function AmActionExec(strFileName As String, strDirectory As String, strParameters As String, strTable As String, IRecordId As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	<input checked="" type="checkbox"/>
リンクまたはフィールドの設定スクリプト	<input checked="" type="checkbox"/>
「スクリプト」タイプのアクションの設定	<input checked="" type="checkbox"/>
ウィザードスクリプト	<input checked="" type="checkbox"/>
ウィザードの「FINISH.DO」スクリプト	<input checked="" type="checkbox"/>

## パラメータ

- strFileName : 実行可能ファイルまたは文書（ファイルマネージャで実行可能ファイルとして設定されている場合）の名前
- strDirectory : strFileNameに指定されているファイルへのパス。
- strParameters : 実行可能ファイルの起動時に実行可能ファイルに提供する各種のパラメータ（必要な場合のみ）
- strTable : アクションのコンテキストを指定する時に追加するパラメータ。アクションを適用するレコードが入っているテーブルのSQL名を指定します。

- IRecordId : アクションのコンテキストを指定する時に追加するパラメータ。アクションを適用するレコードのIDを指定します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## 例

「C」ドライブの「WinNT」ディレクトリにあるWindows NT Explorerを実行します。

```
RetVal = AmActionExec("explorer.exe", "c:/winnt/")
```

## AmActionMail()

AssetCenterが管理する以下のメッセージシステムのうちのいずれか1つを介してメッセージを送信します。

- 内部メッセージシステム
- VIM規格をベースとする外部のメッセージシステム ( Lotus Notesなど )
- MAPI規格をベースとする外部のメッセージシステム ( Microsoft Exchange、Microsoft Outlookなど )
- SMTP規格をベースとする外部のメッセージシステム ( インターネット規格 )

## APIシンタックス

```
long AmActionMail(char *strTo, char *strCc, char *strCcc, char *strSubject, char *strMessage, long iPriority, long bAcknowledge, char *strRefObject, char *strTable, long IRecordId);
```

## 内部BASICシンタックス

```
Function AmActionMail(strTo As String, strCc As String, strCcc As String, strSubject As String, strMessage As String, iPriority As Long, bAcknowledge As Long, strRefObject As String, strTable As String, IRecordId As Long) As Long
```

## 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strTo：このパラメータには、メッセージの受取人のアドレスのリストが、「メッセージシステム：アドレス」の形で列挙されています。セミコロンが区切り文字として使用されています。
- strCc：メッセージのCarbon Copyを送信するアドレスのリスト。アドレスをコンマ(;)で区切ります。
- strCcc：メッセージのBlind Carbon Copyを送信するアドレスのリスト（受取人のリストには表示されません）。アドレスをコンマで区切ります。
- strSubject：メッセージの件名
- strMessage：メッセージの本文
- iPriority：メッセージ送信の優先度を指定します。
  - 0：低
  - 1：普通
  - 2：高
- bAcknowledge：メッセージの送信者が、受信者から通知を受け取るかどうかを指定します。
  - 0：送信者は通知を受け取りません。
  - 1：送信者は通知を受け取ります。
- strRefObject：AssetCenterの内部メッセージシステムを介して送信されたメッセージにのみ使用します。参照されたオブジェクトに達するため、実行コンテキストに対応するレコードから追跡するリンクのSQL名が、このパラメータには含まれています。CurrentUser仮想リンクを使用することも可能です。
- strTable：アクションのコンテキストを指定する時に追加するパラメータ。アクションを適用するレコードが入っているテーブルのSQL名を指定します。
- IRecordId：アクションのコンテキストを指定する時に追加するパラメータ。アクションを適用するレコードのIDを指定します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmActionPrint()

データベースで指定したレコードのレポートを印刷します。

## 内部BASICシンタックス

Function AmActionPrint(IReportId As Long, IRecordId As Long) As Long

## 用途

バージョン : 3.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IReportId : 印刷するレポートのID
- IRecordId : レポートを印刷するレコードのID。デフォルトでは、このパラメータは「0」に設定されています。関連するテーブルはレポートが暗黙に定義しています。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmActionPrintPreview()

データベースのレコードに関するレポートの印刷プレビューを起動します。

### 内部BASICシンタックス

Function AmActionPrintPreview(IReportId As Long, IRecordId As Long) As Long

### 用途

バージョン : 3.60

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- IReportId : 関連するレポートのID。
- IRecordId : レポートに関するレコードの識別子。このパラメータのデフォルト値は「0」です。

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmActionPrintTo()

データベースレコードと、プリンタについてのレポートを印刷します。

## 内部BASICシンタックス

Function AmActionPrintTo(strPrinterName As String, IReportId As Long, IRecordId As Long) As Long

## 用途

バージョン : 3.5

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strPrinterName : 印刷を実行するプリンタの名前。
- IReportId : 印刷するレポートのID
- IRecordId : レポートに関するレコードのID。このパラメータのデフォルト値は「0」です。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmAddAllPOLinesToInv()

既存の請求書に1件の発注を追加します。

## APIシンタックス

long AmAddAllPOLinesToInv(long hApiCnxBase, long IPOrdId, long IInvId);

## 内部BASICシンタックス

Function AmAddAllPOLinesToInv(IPOrdId As Long, IInvId As Long) As Long

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IPOrdId：請求書に追加する発注のID
- IInvId：発注を追加する請求書のID

### 戻りコード

- 0：成功
- 0以外：エラーコード

## AmAddCatRefAndCompositionToPOrder()

ある発注にカタログリファレンスの内容を全部追加します。

### APIシンタックス

```
long AmAddCatRefAndCompositionToPOrder(long hApiCnxBase, long IPOrdId, long ICatRefId, float fCatRefQty, long IRequestId, double dUnitPrice, char *strCur);
```

## 内部BASICシンタックス

Function AmAddCatRefAndCompositionToPOOrder(IPOrderId As Long, ICatRefId As Long, fCatRefQty As Single, IRequestId As Long, dUnitPrice As Double, strCur As String) As Long

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IPOrderId：記入する発注のID
- ICatRefId：カタログリファレンスのID
- fCatRefQty：追加する数量（製品に関連付けられた単位で）
- IRequestId：この発注の元となる依頼のID
- dUnitPrice：カタログリファレンスの製品の単価
- strCur：単価を表記するための通貨のコード

## 戻りコード

- 0：成功
- 0以外：エラーコード

## 注

 注意:

発注に記入するために、カタログリファレンスの製品構成を使用します。

## AmAddCatRefToPOrder()

カタログリファレンスを既存の発注に追加します。

### APIシンタックス

```
long AmAddCatRefToPOrder(long hApiCnxBase, long IRequestLineId, long ICatRefId,
long IPOrderId, float fQty, long bCanMerge);
```

### 内部BASICシンタックス

```
Function AmAddCatRefToPOrder(IRequestLineId As Long, ICatRefId As Long, IPOrderId
As Long, fQty As Single, bCanMerge As Long) As Long
```

### 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IRequestLineId : 発注に関連する依頼明細のID
- ICatRefId : 追加するカタログリファレンスのID
- IPOrderId : 演算の適用先の発注のID
- fQty : 追加する数量 (製品に関連付けられた単位で)
- bCanMerge : 追加明細を発注内に既存する明細と結合するかどうかを指定します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmAddEstimLinesToPO()

既存の発注に、1件の見積の全見積明細を追加します。

## APIシンタックス

```
long AmAddEstimLinesToPO(long hApiCnxBase, long IEstimId, long IPOrdId, long bMergeLines);
```

## 内部BASICシンタックス

```
Function AmAddEstimLinesToPO(IEstimId As Long, IPOrdId As Long, bMergeLines As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IEstimId : 発注に追加する見積のID
- IPOrdId : 1件の見積の全見積明細を追加する発注のID
- bMergeLines : 同じ依頼明細 ( bMergeLines=1 ) を1つの明細に統合するかどうかを指定できます。統合する明細の数量を合計して、1つの明細を作成します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmAddEstimLineToPO()

既存の発注に、1個の見積明細を追加します。

## APIシンタックス

```
long AmAddEstimLineToPO(long hApiCnxBase, long IEstimLineId, long IPOrdId, long bMergeLines);
```

## 内部BASICシンタックス

```
Function AmAddEstimLineToPO(IEstimLineId As Long, IPOrdId As Long, bMergeLines As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	

使用

ウィザードの「FINISH.DO」スクリプト



## パラメータ

- IEstimLineId : 発注に追加する見積明細のID
- IPOrdId : 見積明細を追加する発注のID
- bMergeLines : 同じ依頼明細 ( bMergeLines=1 ) を1つの明細に統合するかどうかを指定できます。統合する明細の数量を合計して、1つの明細を作成します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmAddPOLineToInv()

1つの発注明細の数量のうち、指定した数量だけを1件の請求書に追加します。

## APIシンタックス

```
long AmAddPOLineToInv(long hApiCnxBase, long IPOrdLineId, long IInvId, float fQty);
```

## 内部BASICシンタックス

```
Function AmAddPOLineToInv(IPOrdLineId As Long, IInvId As Long, fQty As Single)  
As Long
```

## 用途

バージョン : 3.00

使用

AssetCenter API



## 使用

---

 リンクまたはフィールドの設定スクリプト
 

---

 「スクリプト」タイプのアクションの設定 


---

 ウィザードスクリプト
 

---

 ウィザードの「FINISH.DO」スクリプト 


---

## パラメータ

- IPOrdLineId : 請求書に追加する発注明細のID
- IInvid : 発注明細の物件を追加する請求書のID
- fQty : 請求書に追加する発注明細の物件の数量

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmAddPOrderLineToReceipt()

受領に発注明細を追加します。これにより、既存の受領内で発注明細を受領できます。

## APIシンタックス

```
long AmAddPOrderLineToReceipt(long hApiCnxBase, long IPOrdLineId, long IRecptId, float fQty, long bCanMerge);
```

## 内部BASICシンタックス

```
Function AmAddPOrderLineToReceipt(IPOrdLineId As Long, IRecptId As Long, fQty As Single, bCanMerge As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IOrderLineId：発注明細のID
- IRecptId：影響を受ける受領のID
- fQty：受領する数量。発注した数量の内、受領する数量を指定できます（製品の単位で）。
- bCanMerge：このパラメータでは、受領内に既存の明細に明細を結合するかどうかを指定できます。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmAddReceiptLineToInvoice()

請求書に受領明細を追加します。これにより、既存の請求書内で受領明細を請求できます。

## APIシンタックス

```
long AmAddReceiptLineToInvoice(long hApiCnxBase, long IRecptLineId, long IInvoiceId,
float fQty, long bCanMerge);
```

## 内部BASICシンタックス

```
Function AmAddReceiptLineToInvoice(IRecptLineId As Long, IInvoiceId As Long, fQty
As Single, bCanMerge As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IRecptLineId：受領明細のID
- IInvoiceId：影響を受ける請求書のID
- fQty：請求する数量。受領した数量の内、請求する数量を指定できます（製品の単位で）。
- bCanMerge：このパラメータでは、請求書の既存明細に明細を結合するかどうかを指定できます。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmAddReqLinesToEstim()

既存の見積に1件の依頼の全依頼明細を追加します。

### APIシンタックス

```
long AmAddReqLinesToEstim(long hApiCnxBase, long IReqId, long IEstimId, long bMergeLines);
```

### 内部BASICシンタックス

```
Function AmAddReqLinesToEstim(IReqId As Long, IEstimId As Long, bMergeLines As Long) As Long
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IReqId：見積に追加する依頼のID
- IEstimId：依頼の全依頼明細を追加する見積のID
- bMergeLines：同じ依頼明細（bMergeLines=1）を1つの明細に統合するかどうかを指定できます。統合する明細の数量を合計して、1つの明細を作成します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmAddReqLinesToPO()

既存の発注に1件の依頼のすべての依頼明細を追加します。依頼で指定するサプライヤと発注で指定するサプライヤが同じでなければなりません。

## APIシンタックス

```
long AmAddReqLinesToPO(long hApiCnxBase, long IReqId, long IPOrdId, long bMergeLines);
```

## 内部BASICシンタックス

```
Function AmAddReqLinesToPO(IReqId As Long, IPOrdId As Long, bMergeLines As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IReqId : 発注に追加する依頼のID
- IPOrdId : 依頼明細を追加する発注のID

- bMergeLines : 同じ依頼明細 ( bMergeLines=1 ) を1つの明細に統合するかどうかを指定できます。統合する明細の数量を合計して、1つの明細を作成します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmAddReqLineToEstim()

既存の見積りに1つの依頼明細を追加します。

## APIシンタックス

```
long AmAddReqLineToEstim(long hApiCnxBase, long lReqLineId, long lEstimId, long bMergeLines);
```

## 内部BASICシンタックス

```
Function AmAddReqLineToEstim(lReqLineId As Long, lEstimId As Long, bMergeLines As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IReqLineId : 見積に追加する依頼明細のID
- IEstimId : 依頼明細を追加する見積のID
- bMergeLines : 同じ依頼明細 ( bMergeLines=1 ) を1つの明細に統合するかどうかを指定できます。統合する明細の数量を合計して、1つの明細を作成します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmAddReqLineToPO()

既存の発注に1つの依頼明細を追加します。

## APIシンタックス

```
long AmAddReqLineToPO(long hApiCnxBase, long IReqLineId, long IPOrdId, long bMergeLines);
```

## 内部BASICシンタックス

```
Function AmAddReqLineToPO(IReqLineId As Long, IPOrdId As Long, bMergeLines As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	

使用

ウィザードの「FINISH.DO」スクリプト



## パラメータ

- IReqLineId : 発注に追加する依頼明細のID
- IPOrdId : 依頼明細を追加する発注のID
- bMergeLines : 同じ依頼明細 ( bMergeLines=1 ) を1つの明細に統合するかどうかを指定できます。統合する明細の数量を合計して、1つの明細を作成します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmAddRequestLineToPOrder()

発注に1つの依頼明細を追加します。

## APIシンタックス

```
long AmAddRequestLineToPOrder(long hApiCnxBase, long IRequestLineId, long IPOrdId, float fQty, long bCanMerge);
```

## 内部BASICシンタックス

```
Function AmAddRequestLineToPOrder(IRequestLineId As Long, IPOrdId As Long, fQty As Single, bCanMerge As Long) As Long
```

## 用途

バージョン : 4.00

## 使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IRequestId : 依頼明細のID
- IOrderId : 影響を受ける発注のID
- fQty : 発注する数量。依頼された数量の内、発注する数量を指定できます（モデルの単位で）。
- bCanMerge : このパラメータでは、発注の既存明細に明細を結合するかどうかを指定できます。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmAddTemplateToPOrder()

ある発注に発注テンプレートの内容を全部追加します。

## APIシンタックス

```
long AmAddTemplateToPOrder(long hApiCnxBase, long IRequestId, long IOrderId,
long ITemplateId, long IQty, long bCanMerge);
```

## 内部BASICシンタックス

Function AmAddTemplateToPOOrder(IRequestId As Long, IPOrderId As Long, ITemplateId As Long, IQty As Long, bCanMerge As Long) As Long

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IRequestId：追加される発注明細の元となる依頼明細のID
- IPOrderId：影響を受ける発注のID
- ITemplateId：追加する発注テンプレートのID
- IQty：追加する数量（製品の単位で）
- bCanMerge：このパラメータでは、発注の既存明細に明細を結合するかどうかを指定できます。

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmAddTemplateToRequest()

ある依頼に依頼テンプレートの内容を全部追加します。

## APIシンタックス

```
long AmAddTemplateToRequest(long hApiCnxBase, long IReqId, long ITemplateId,
long IQty, long bCanMerge);
```

## 内部BASICシンタックス

```
Function AmAddTemplateToRequest(IReqId As Long, ITemplateId As Long, IQty As
Long, bCanMerge As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IReqId：影響を受ける依頼明細のID
- ITemplateId：追加する依頼テンプレートのID
- IQty：追加する数量（製品の単位で）
- bCanMerge：このパラメータでは、依頼の既存明細に明細を結合するかどうかを指定できます。

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmBusinessSecondsInDay()

カレンダーに従って、1日の業務時間を秒単位で計算します。

### APIシンタックス

```
long AmBusinessSecondsInDay(char *strCalendarSqlName, long tmDate);
```

### 内部BASICシンタックス

```
Function AmBusinessSecondsInDay(strCalendarSqlName As String, tmDate As Date)
As Date
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strCalendarSqlName : 計算に使用するカレンダーのSQL名
- tmDate : 計算を実行する日付

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCalcConsolidatedFeature()

SQL名でテーブルを識別し、そのテーブルに含まれる合計された任意管理項目の値を計算します。

### APIシンタックス

```
long AmCalcConsolidatedFeature(long hApiCnxBase, long ICalcFeatId, char *strSQLTableName);
```

### 内部BASICシンタックス

```
Function AmCalcConsolidatedFeature(ICalcFeatId As Long, strSQLTableName As String) As Long
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- ICalcFeatId：合計された任意管理項目のID
- strSQLTableName：合計された任意管理項目を計算するテーブルのSQL名。任意管理項目は、このテーブルの任意管理項目でなければなりません。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCalcDepr()

指定した日付の資産の減価償却の金額を計算できます。この日付の減価償却の値を返します。

## APIシンタックス

```
double AmCalcDepr(long iType, long IDuration, double dCoeff, double dPrice, long tmStart, long tmDate);
```

## 内部BASICシンタックス

```
Function AmCalcDepr(iType As Long, IDuration As Long, dCoeff As Double, dPrice As Double, tmStart As Date, tmDate As Date) As Double
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- iType : 減価償却の属性を指定します。次のいずれかの値になります。
  - 0 : 減価償却なし
  - 1 : 定額法

- 2：定率法
- IDuration：資産が減価償却される期間。この期間は秒単位で示されます。
- dCoeff：定率法の減価償却に適用する償却率。定額法ではこのパラメータは使用されませんが、値は設定しておく必要があります。
- dPrice：減価償却を計算する資産の初期価額
- tmStart：資産の減価償却を開始する日付
- tmDate：資産の減価償却と残存価額を計算する日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCbkReplayEvent()

この関数はイベントの起点にあるレコードを訂正した後、イベントの起点で経費付替え規則を再適用します。

## APIシンタックス

```
long AmCbkReplayEvent(long hApiCnxBase, long ICbkEventId);
```

## 内部BASICシンタックス

```
Function AmCbkReplayEvent(ICbkEventId As Long) As Long
```

## 用途

バージョン：4.00

---

AssetCenter API	使用
-----------------	----

---

## 使用

---

 リンクまたはフィールドの設定スクリプト

 「スクリプト」タイプアクションの設定 


---

ウィザードスクリプト

 ウィザードの「FINISH.DO」スクリプト 


---

## パラメータ

- ICbkEventId : 関連する経費付替えイベントのID

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCheckTraceDone()

AmCheckTraceDone APIは、ポート (IPortId) またはバンドル (IBundleId) が既存のトレースに関連付けられるかを決めます。トレースの方向 (iTraceDir) はトレースが、ユーザからホストへの方向 (iTraceDir=1) で検証されるべきか、またはホストからユーザへの方向 (iTraceDir=0) で検証されるべきかを指定します。

## APIシンタックス

```
long AmCheckTraceDone(long hApiCnxBase, long IPortId, long IBundleId, long iTraceDir);
```

## 内部BASICシンタックス

```
Function AmCheckTraceDone(IPortId As Long, IBundleId As Long, iTraceDir As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IPortId : 確認するポートのID
- IBundleId : 確認するバンドルのID
- iTraceDir : このパラメータは検証する方向を指定します。
  - 1 : ホストの方向で検証する
  - 0 : ユーザの方向で検証する

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCleanup()

この関数は、データベースの変更関数を使うスクリプトの最後に呼び出されません。使用したすべてのリソースを解放します。

## APIシンタックス

```
void AmCleanup();
```

## 用途

バージョン : 2.52

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## AmClearLastError()

最後に関数を呼び出した際に発生した最後のエラーメッセージに関する情報を消去します。

## APIシンタックス

```
long AmClearLastError(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmClearLastError() As Long
```

## 用途

バージョン : 2.52

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCloseAllChildren()

現在の接続で作成されたすべてのオブジェクトを破棄します。

### APIシンタックス

```
long AmCloseAllChildren(long hApiCnxBase);
```

### 内部BASICシンタックス

```
Function AmCloseAllChildren() As Long
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCloseConnection()

指定した接続のAssetCenterセッションを終了します。接続中に作成したすべてのオブジェクト（クエリ、レコード、テーブル、フィールドなど）は自動的に破棄されます。ハンドルは無効になります。接続ハンドルは存在しません。

## APIシンタックス

long AmCloseConnection(long hApiCnxBase);

## 用途

バージョン : 2.52

使用

AssetCenter API



リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCommit()

接続に関連付けられているデータベースに加えられたすべての変更を有効にします。

## APIシンタックス

long AmCommit(long hApiCnxBase);

## 内部BASICシンタックス

Function AmCommit() As Long

## 用途

バージョン : 2.52

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmComputeAllLicAndInstallCounts()

すべてのレコードについて、ソフトウェアライセンス数とインストール数をカウントします。

## APIシンタックス

```
long AmComputeAllLicAndInstallCounts(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmComputeAllLicAndInstallCounts() As Long
```

## 用途

バージョン : 3.00

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmComputeLicAndInstallCounts()

1つのレコードについて、ソフトウェアライセンス数とインストール数をカウントします。

## APIシンタックス

```
long AmComputeLicAndInstallCounts(long hApiCnxBase, long ISLCountId);
```

## 内部BASICシンタックス

```
Function AmComputeLicAndInstallCounts(ISLCountId As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- ISLCountId : ソフトウェアライセンスカウンタのID

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmConnectTrace()

AmConnectTrace APIはソースデバイス/ケーブルをターゲットデバイス/ケーブルへ接続し、トレース履歴とトレースの処理を作成します。

## APIシンタックス

```
long AmConnectTrace(long hApiCnxBase, long iSrcLinkType, long ISrcPortBunId, long ISrcLabelRuleId, long iDestLinkType, long IDestPortBunId, long IDestLabelRuleId, long iTraceDir, long IDutyId, char *strComment, long ICabTraceOutId);
```

## 内部BASICシンタックス

```
Function AmConnectTrace(iSrcLinkType As Long, ISrcPortBunId As Long, ISrcLabelRuleId As Long, iDestLinkType As Long, IDestPortBunId As Long, IDestLabelRuleId As Long, iTraceDir As Long, IDutyId As Long, strComment As String, ICabTraceOutId As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- iSrcLinkType : このパラメータはソースデバイス/ケーブル用のトレースのタイプを決めます。
  - 8 : ケーブル
  - 9 : デバイス
- iSrcPortBunId : ソース側で接続するポートまたはバンドルのID
- iSrcLabelRuleId : ソースケーブルリンク用のラベル付け規則のID
- iDestLinkType : このパラメータはターゲットデバイス/ケーブル用のトレースのタイプを決めます。
  - 8 : ケーブル
  - 9 : デバイス
- iDestPortBunId : ターゲット側で接続するポートまたはバンドルのID
- iDestLabelRuleId : ターゲットケーブルリンク用のラベル付け規則のID
- iTraceDir : このパラメータは接続の方向を指定します。
  - 1 : ユーザからホストへ
  - 0 : ホストからユーザへ
- iDutyId : ケーブルタイプのリンクの用途ID
- strComment : トレースの処理のラベル
- iCabTraceOutId : ケーブルのトレースの説明のID

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertCurrency()

指定した日付の金額を、別の通貨に換算します。

## APIシンタックス

```
double AmConvertCurrency(long hApiCnxBase, long tmDate, char *strSrcName, char *strDstName, double dVal);
```

## 内部BASICシンタックス

```
Function AmConvertCurrency(tmDate As Date, strSrcName As String, strDstName As String, dVal As Double) As Double
```

## 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- tmDate：換算日。この日付現在の有効な為替レートを取得するために使います。
- strSrcName：換算する金額の元の通貨
- strDstName：換算通貨。金額の換算に使う通貨です。
- dVal：換算する金額（元の通貨による金額）

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注



### 注意:

この関数の通貨パラメータ (strSrcNameとstrDstName) は、必ずAssetCenterで定義しておく必要があります。また、換算日 (tmDateパラメータ) 現在の有効な為替レートがなければなりません。

## 例

1998年11月2日に、5,000フランをドルに換算します。

```
AmConvertCurrency("1998/11/02 00:00:00", "FRF", "$", 5000)
```

## AmConvertDateBasicToUnix()

BASIC形式の日付 (Date型) をUNIX形式の日付 (Long型) に変換します。外部ツールからこの関数を使用しても、2つのタイプは同等なため機能しません。

## APIシンタックス

```
long AmConvertDateBasicToUnix(long hApiCnxBase, long tmTime);
```

## 内部BASICシンタックス

```
Function AmConvertDateBasicToUnix(tmTime As Date) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	

使用

ウィザードの「FINISH.DO」スクリプト



## パラメータ

- tmTime : 変換する日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertDateIntlToUnix()

国際標準形式の日付（Date型）をUNIX形式の日付（Long型）に変換します。

## APIシンタックス

```
long AmConvertDateIntlToUnix(long hApiCnxBase, char *strDate);
```

## 内部BASICシンタックス

```
Function AmConvertDateIntlToUnix(strDate As String) As Long
```

## 用途

バージョン : 3.00

使用

AssetCenter API



リンクまたはフィールドの設定スクリプト



	使用
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strDate : 国際標準形式 ( yyyy-mm-dd hh:mm:ss ) 変換する日付

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertDateToStringToUnix()

文字列の日付（Windowsのコントロールパネルに表示される形式）をUNIXのLong型に変換します。

## APIシンタックス

```
long AmConvertDateToStringToUnix(long hApiCnxBase, char *strDate);
```

## 内部BASICシンタックス

```
Function AmConvertDateToStringToUnix(strDate As String) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strDate : 変換する文字列の日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertDateUnixToBasic()

UNIX形式の日付（Long型）をBASIC形式の日付（Date型）に変換します。外部ツールからこの関数を使用しても、2つのタイプは同等なため機能しません。

## APIシンタックス

```
long AmConvertDateUnixToBasic(long hApiCnxBase, long lTime);
```

## 内部BASICシンタックス

```
Function AmConvertDateUnixToBasic(lTime As Long) As Date
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- ITime : 変換する日付

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertDateUnixToIntl()

UNIX形式の日付（Long型）を国際標準形式の日付（yyyy-mm-dd hh:mm:ss）に変換します。

## APIシンタックス

```
long AmConvertDateUnixToIntl(long hApiCnxBase, long lUnixDate, char *pstrDate, long lDate);
```

## 内部BASICシンタックス

Function AmConvertDateUnixToIntl(IUnixDate As Long) As String

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IUnixDate：変換する日付

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertDateUnixToString()

Long型のUNIX形式の日付を文字列形式の日付（Windowsのコントロールパネルに表示される形式）に変換します。

## APIシンタックス

```
long AmConvertDateUnixToString(long hApiCnxBase, long IUnixDate, char *pstrDate,
long IDate);
```

## 内部BASICシンタックス

```
Function AmConvertDateUnixToString(IUnixDate As Long) As String
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IUnixDate : 変換するLong型のUNIX形式の日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertDoubleToString()

倍精度の数字を文字列に変換します。文字列は、Windowsのコントロールパネルで設定した地域のオプション（数字）に応じてフォーマットされます。

### APIシンタックス

```
long AmConvertDoubleToString(double dSrc, char *pstrDst, long lDst);
```

### 内部BASICシンタックス

```
Function AmConvertDoubleToString(dSrc As Double) As String
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- dSrc : 変換する倍精度の数字

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertMonetaryToString()

金額値を文字列に変換します。文字列は、Windowsのコントロールパネルで設定した地域のオプション（通貨）に応じてフォーマットされます。

### APIシンタックス

```
long AmConvertMonetaryToString(double dSrc, char *pstrDst, long lDSt);
```

### 内部BASICシンタックス

```
Function AmConvertMonetaryToString(dSrc As Double) As String
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- dSrc：変換する金額値

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertStringToDouble()

文字列（Windowsのコントロールパネルで設定したフォーマット）を倍精度の数字に変換します。

### APIシンタックス

```
double AmConvertStringToDouble(char *strSrc);
```

### 内部BASICシンタックス

```
Function AmConvertStringToDouble(strSrc As String) As Double
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strSrc：変換する文字列

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmConvertStringToMonetary()

文字列（Windowsのコントロールパネルで設定したフォーマット）を金額値に変換します。

### APIシンタックス

```
double AmConvertStringToMonetary(char *strSrc);
```

### 内部BASICシンタックス

```
Function AmConvertStringToMonetary(strSrc As String) As Double
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strSrc：変換する文字列

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCounter()

strCounterNameカウンタの値よりも1つ大きい値を返します。iWidthがカウンタの桁数よりも大きい場合は、ゼロが先頭に追加されます。iWidthの値よりもカウンタの桁数の方が大きい場合は、関数が返す結果は決して切り捨てられません。

## 内部BASICシンタックス

Function AmCounter(strCounterName As String, iWidth As Long) As String

## 用途

バージョン : 2.52

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト 

「スクリプト」タイプのアクションの設定

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト

---

## パラメータ

- strCounterName : AssetCenterで定義されているカウンタの名前 ( [ 管理 / カウンタ ] メニューからアクセスできます。 )
- iWidth : この関数の出力を何桁で表示するかを指定します。このパラメータは、カウンタのサイズがパラメータ値よりも小さい場合のみ使用されます。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数 ( 必要に応じてAmLastErrorMsg()関数も ) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

「納入」カウンタの値を5桁で返します。

```
Dim strCounterName As String
strCounter = AmCounter("納入", 5)
```

例えば、「納入」カウンタが18の場合は次の値を返します。

```
00019
```

## AmCreateAssetPort()

AmCreateAssetPort APIはデバイス ( IAssetId ) 用に新規のポートを作成します。新規のポートには、あるケーブルコネクタタイプ ( ICabCnxTypeId ) のピン ( iPinCount ) が付属しています。ピンのステータスは「使用可能」でなければなりません。ポートに追加されるピンは連続番号で並べ替えられます。ポートの方向 ( bPinPortDir ) に従って、使用可能なピンは昇順 ( bPinPortDir=0 ) または降順 ( bPinPortDir=1 ) で並べ替えられます。この関数は新規のポートに用途 ( IDutyId ) を割り当てます。

## APIシンタックス

```
long AmCreateAssetPort(long hApiCnxBase, long IAssetId, long ICabCnxTypeId, long IDutyId, long iPinCount, long bPinPortDir, long iConnStatus, long bConsecutivePins, long iPrevPinSeq, long bLogError);
```

## 内部BASICシンタックス

```
Function AmCreateAssetPort(IAssetId As Long, ICabCnxTypeId As Long, IDutyId As Long, iPinCount As Long, bPinPortDir As Long, iConnStatus As Long, bConsecutivePins As Long, iPrevPinSeq As Long, bLogError As Long) As Long
```

## 用途

バージョン : 4.00

使用

AssetCenter API



## 使用

---

 リンクまたはフィールドの設定スクリプト
 

---

 「スクリプト」タイプのアクションの設定 


---

 ウィザードスクリプト
 

---

 ウィザードの「FINISH.DO」スクリプト 


---

## パラメータ

- IAssetId : デバイスのID
- ICabCnxTypeId : ケーブルの接続タイプのID
- IDutyId : ポートの用途タイプのID
- iPinCount : 新規のポートで使用されるピンの数
- bPinPortDir : このパラメータはポートの方向を指定します。
- iConnStatus
- bConsecutivePins
- iPrevPinSeq
- bLogError

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateAssetsAwaitingDelivery()

この関数は納品待ちの資産を作成します。

## APIシンタックス

```
long AmCreateAssetsAwaitingDelivery(long hApiCnxBase, long IPOrdId);
```

## 内部BASICシンタックス

Function AmCreateAssetsAwaitingDelivery(IPOrdId As Long) As Long

### 用途

バージョン : 3.61

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IPOrdId : 関連する発注のID

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCreateCable()

AmCreateCable APIは新規ケーブルを作成します。ケーブルはモデル ( IModelId )、ケーブルの役割 ( strCableRole )、ラベル付け規則 ( ILabelRuleId )、ユーザの場所 ( IUserId ) とホストの場所 ( IHostId ) に基づいて作成されます。プロジェクト ( IProjectId ) と作業指示 ( IWorkOrderId ) に値が入力されると、新規のケーブルは、コメント ( strComment ) と共にプロジェクトや作業指示に追加されます。このコメントはケーブル上に実行されるアクションを説明します ( 例 : 新規ケーブルの設置 )。

## APIシンタックス

```
long AmCreateCable(long hApiCnxBase, long IModelId, long IUserId, long IHostId, char
*strCableRole, long IProjectId, long IWorkOrderId, char *strComment, long ILabelRuleId,
char *strLabel);
```

## 内部BASICシンタックス

```
Function AmCreateCable(IModelId As Long, IUserId As Long, IHostId As Long,
strCableRole As String, IProjectId As Long, IWorkOrderId As Long, strComment As
String, ILabelRuleId As Long, strLabel As String) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IModelId：ケーブルのモデルのID
- IUserId：ユーザ側の場所のID
- IHostId：ホスト側の場所のID
- strCableRole：ケーブルの役割
- IProjectId：このパラメータはケーブルの設置に関連するプロジェクトを定義します。
- IWorkOrderId：ケーブル設置に関連する作業指示のID
- strComment：作業指示（IWorkOrderIdで指定）に添付されるコメント
- ILabelRuleId：ケーブル用のラベルの作成時に適用されるラベル付け規則のID
- strLabel：このパラメータはケーブルに貼るラベルを指定します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateCableBundle()

AmCreateCableBundle APIはケーブル (ICableId) に新規バンドルを作成します。新規バンドルには、あるケーブルペアタイプ (IPairType) のケーブルペア (iPairCount) が付属しています。ペアのステータスは「使用可能」でなければなりません。この関数は用途 (IDutyId) を新規バンドルに割り当てます。

## APIシンタックス

```
long AmCreateCableBundle(long hApiCnxBase, long ICableId, long IPairTypeId, long IDutyId, long iPairCount, long iStartPairSeq, long bLogError);
```

## 内部BASICシンタックス

```
Function AmCreateCableBundle(ICableId As Long, IPairTypeId As Long, IDutyId As Long, iPairCount As Long, iStartPairSeq As Long, bLogError As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- ICableId : ケーブルのID (このIDはケーブルのテーブルに存在しなければなりません)。
- IPairTypeId : ケーブルペアタイプのID
- IDutyId : バンドルの用途のID
- iPairCount : このパラメータはバンドルのペア数を定義します。
- iStartPairSeq
- bLogError

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数 (必要に応じてAmLastErrorMsg()関数も) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateCableLink()

AmCreateCableLink APIはケーブル (ICableId) とバンドル (IBundleId) 用に新規のケーブルリンクを作成します。ケーブルリンクの用途はある機能 (IDutyId) から設定されます。ケーブルリンクのラベル付け規則は、あるラベル付け規則 (ILabelRuleId) に基づいて設定されます。

### 注意:

ラベル付けがあるラベル付け規則により更新されることはありません。これにはAmRefreshLabel()関数を別に呼び出す必要があります。

以前のケーブルリンク (IPrevLinkId) が指定されている場合、親ケーブルリンクが2レコード間に作成されます。以前のケーブルリンクは、これら2つのレコードの子ケーブルリンクになります。

## APIシンタックス

```
long AmCreateCableLink(long hApiCnxBase, long ICableId, long IDutyId, long IBundleId, long IPrevLinkId, long iTraceDir, long ILabelRuleId);
```

## 内部BASICシンタックス

Function AmCreateCableLink(ICableId As Long, IDutyId As Long, IBundleId As Long, IPrevLinkId As Long, iTraceDir As Long, ILabelRuleId As Long) As Long

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- ICableId：接続用ケーブルID
- IDutyId：接続の機能のID
- IBundleId：接続するケーブルバンドルのID
- IPrevLinkId：接続に使用されるケーブルリンクのID。値「0」を使用するとこのパラメータは必須ではなくなります。
- iTraceDir：接続の方向
  - 0=ホストからユーザへの方向
  - 1=ユーザからホストへの方向
- ILabelRuleId：使用されるラベル付け規則のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateDelivFromPO()

発注した製品を受領し、それによって作成された受領伝票のIDを返します。

### APIシンタックス

```
long AmCreateDelivFromPO(long hApiCnxBase, long IPOrdId);
```

### 内部BASICシンタックス

```
Function AmCreateDelivFromPO(IPOrdId As Long) As Long
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト 「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IPOrdId : 受領する注文品（発注）のID

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateDevice()

AmCreateDevice APIは新規デバイスを作成します。デバイスはモデル ( IModelId ) と場所 ( ILocationId ) を基に作成されます。ラベル付け規則は、ある規則 ( ILabelRuleId ) に応じて設定されます。

### 注意:

ラベル付け規則はラベルを更新しません。更新にはAmRefreshLabel()を別に呼び出す必要があります。

プロジェクト ( IProjectId ) と作業指示 ( IWorkOrderId ) に値が入力されると、新規資産はstrComment内のコメントと共にプロジェクトと作業指示に追加されます。コメントは資産に実行されるアクションを説明します ( 例 : 新規資産の設置 ) 。

## APIシンタックス

```
long AmCreateDevice(long hApiCnxBase, long IModelId, long ILocationId, long IProjectId, long IWorkOrderId, long ILabelRuleId, char *strComment);
```

## 内部BASICシンタックス

```
Function AmCreateDevice(IModelId As Long, ILocationId As Long, IProjectId As Long, IWorkOrderId As Long, ILabelRuleId As Long, strComment As String) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IModelId : 新規デバイスのモデルのID
- ILocationId : 新規デバイスの場所のID
- IProjectId : プロジェクトのID。「0」の値をとることも可能です。
- IWorkOrderId : 作業指示のID。「0」の値をとることも可能です。
- ILabelRuleId : 資産に使用されるラベル付け規則のID
- strComment : 作業指示に添付されるコメント

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateDeviceLink()

AmCreateDeviceLink APIは、あるデバイス (IAssetId) とポート (IPortId) 用にデバイスタイプのケーブルリンクを作成します。ケーブルリンクのラベル付け規則は、あるラベル付け規則 (ILabelRuleId) を元に設定されます。

### 注意:

ラベル付け規則はラベルを更新しません。別にAmRefreshLabel()を呼び出す必要があります。

以前のケーブルリンク (IPrevLinkId) が指定される場合は、親ケーブルリンクが2つのレコード間に作成されます。トレースの方向が「ユーザからホスト」の場合 (iTraceDir=1)、以前のケーブルリンクは子になります。トレースの方向が「ホストからユーザ」の場合 (iTraceDir=0) 以前のケーブルリンクは親になります。

## APIシンタックス

```
long AmCreateDeviceLink(long hApiCnxBase, long IAssetId, long IPortId, long IPrevLinkId, long iTraceDir, long ILabelRuleId);
```

## 内部BASICシンタックス

Function AmCreateDeviceLink(IAssetId As Long, IPortId As Long, IPrevLinkId As Long, iTraceDir As Long, ILabelRuleId As Long) As Long

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IAssetId：接続される資産のID
- IPortId：接続予定のポートのID
- IPrevLinkId：接続を可能にするデバイスのケーブルリンクのID
- iTraceDir：接続の方向
  - 0=ホストからユーザへの方向
  - 1=ユーザからホストへの方向
- ILabelRuleId：新規接続に使用されるラベル付け規則のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateEstimFromReq()

購入依頼から見積を作成して、その見積のIDを返します。

### APIシンタックス

```
long AmCreateEstimFromReq(long hApiCnxBase, long IReqId, long ISuppld);
```

### 内部BASICシンタックス

```
Function AmCreateEstimFromReq(IReqId As Long, ISuppld As Long) As Long
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IReqId : 見積の作成に使用する購入依頼のID
- ISuppld : この関数によって作成される見積のサプライヤID

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateEstimsFromAllReqLines()

依頼から見積を作成して、その見積のIDを返します。

### APIシンタックス

```
long AmCreateEstimsFromAllReqLines(long hApiCnxBase, long IReqId, long
bMergeLines, long IDefSuppld);
```

### 内部BASICシンタックス

```
Function AmCreateEstimsFromAllReqLines(IReqId As Long, bMergeLines As Long,
IDefSuppld As Long) As Long
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- IReqId : 見積の作成元となる依頼のID
- bMergeLines : 同じ依頼明細 ( bMergeLines=1 ) を1つの明細に統合するかどうかを指定できます。統合する明細の数量を合計して、1つの明細を作成します。
- IDefSuppld : 見積のデフォルトサプライヤのID

### 戻りコード

- 0 : 成功

- 0以外：エラーコード

## AmCreateInvFromPO()

発注から請求書を作成して、その請求書のIDを返します。

### APIシンタックス

```
long AmCreateInvFromPO(long hApiCnxBase, long IPOrdId);
```

### 内部BASICシンタックス

```
Function AmCreateInvFromPO(IPOrdId As Long) As Long
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IPOrdId：請求書の作成元となる発注のID

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateLink()

レコードのリンクを変更して、ターゲットテーブルの新規レコード（hApiRecDest）を指すようにします。つまり2レコード間のリンクを作成します。

## APIシンタックス

```
long AmCreateLink(long hApiRecord, char *strLinkName, long hApiRecDest);
```

## 内部BASICシンタックス

```
Function AmCreateLink(hApiRecord As Long, strLinkName As String, hApiRecDest As Long) As Long
```

## 用途

バージョン：3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiRecord：変更するリンクを含むレコードのハンドル
- strLinkName：変更するリンクのSQL名
- hApiRecDest：リンクのターゲットレコードのハンドル

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCreatePOFromEstim()

見積から発注を作成して、その発注のIDを返します。

## APIシンタックス

```
long AmCreatePOFromEstim(long hApiCnxBase, long lEstimId);
```

## 内部BASICシンタックス

```
Function AmCreatePOFromEstim(lEstimId As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- lEstimId : 発注の作成に使用する見積のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreatePOFromReq()

購入依頼から発注を作成して、その発注のIDを返します。

### APIシンタックス

```
long AmCreatePOFromReq(long hApiCnxBase, long IReqId, long ISupId);
```

### 内部BASICシンタックス

```
Function AmCreatePOFromReq(IReqId As Long, ISupId As Long) As Long
```

### 用途

バージョン：3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- IReqId：発注の作成に使用する購入依頼のID
- ISupId：作成する発注のサプライヤのID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreatePOrderFromRequest()

依頼から発注を作成します。

## APIシンタックス

```
long AmCreatePOrderFromRequest(long hApiCnxBase, long IRequestId, long ISupplierId);
```

## 内部BASICシンタックス

```
Function AmCreatePOrderFromRequest(IRequestId As Long, ISupplierId As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IRequestId：関連する依頼のID

- ISupplierId : 発注用のサプライヤのID

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreatePOOrdersFromRequest()

依頼に応じるために必要な発注をすべて作成します。

## APIシンタックス

```
long AmCreatePOOrdersFromRequest(long hApiCnxBase, long IRequestId);
```

## 内部BASICシンタックス

```
Function AmCreatePOOrdersFromRequest(IRequestId As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IRequestId : 関連する依頼のID

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCreatePOsFromAllReqLines()

1件の依頼の各依頼明細から発注を作成します。

## APIシンタックス

```
long AmCreatePOsFromAllReqLines(long hApiCnxBase, long IReqId, long bMergeLines,
long IDefSuppld);
```

## 内部BASICシンタックス

```
Function AmCreatePOsFromAllReqLines(IReqId As Long, bMergeLines As Long,
IDefSuppld As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IReqId : 発注を作成する依頼のID
- bMergeLines : 同じ依頼明細 ( bMergeLines=1 ) を1つの明細に統合するかどうかを指定できます。統合する明細の数量を合計して、1つの明細を作成します。
- IDefSuppld : 依頼する製品のデフォルトのサプライヤのID。このパラメータはオプションで、デフォルトの設定は「0」です。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmCreateProjectCable()

AmCreateProjectCable APIは、ケーブル ( ICableId ) をプロジェクト ( IProjectId ) と作業指示 ( IWorkOrderId ) に追加します。コメント ( strComment ) は実行されるアクションを説明します ( 例 : 新規ケーブルの設置 ) 。

## APIシンタックス

```
long AmCreateProjectCable(long hApiCnxBase, long IProjectId, long IWorkOrderId,
long ICableId, char *strComment);
```

## 内部BASICシンタックス

```
Function AmCreateProjectCable(IProjectId As Long, IWorkOrderId As Long, ICableId
As Long, strComment As String) As Long
```

## 用途

バージョン : 4.00

使用

---

AssetCenter API



使用

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定 

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- IProjectId : 新規ケーブルを取得するプロジェクトのID
- IWorkOrderId : ケーブル用の作業指示のID
- ICableId : ケーブルのID
- strComment : 作業指示に添付されるコメント

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateProjectDevice()

AmCreateProjectDevice APIは、デバイス (IAssetId) をプロジェクト (IProjectId) と作業指示 (IWorkOrderId) に追加します。コメント (strComment) は実行されるアクションを説明します (例: 新規デバイスの設置)。

## APIシンタックス

```
long AmCreateProjectDevice(long hApiCnxBase, long IProjectId, long IWorkOrderId,
long IAssetId, char *strComment);
```

## 内部BASICシンタックス

Function AmCreateProjectDevice(IProjectId As Long, IWorkOrderId As Long, IAssetId As Long, strComment As String) As Long

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IProjectId：新規デバイスを取得するプロジェクトのID
- IWorkOrderId：新規デバイスを取得する作業指示のID
- IAssetId：資産としての新規デバイスのID
- strComment：作業指示に添付されるコメント

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateProjectTrace()

AmCreateProjectTrace APIはトレース（strTrace）をプロジェクト（IProjectId）と作業指示（IWorkOrderId）に追加します。トレースの用途は、ある用途（IDutyId）を基に設定されます。トレースのタイプ（iTraceType）は、トレースが接続

(iTraceType = 1) であるか、切断 (iTraceType = 2) であるかを指定します。変更されたユーザリンクのラベル (strModLinkLabel) は、トレースのどの部分が変更されたかを識別します。コメント (strComment) は実行されるアクションを説明します (例: デバイスの接続)。

## APIシンタックス

```
long AmCreateProjectTrace(long hApiCnxBase, long IProjectId, long IWorkOrderId,
long iTraceType, long IDutyId, char *strModLinkLabel, char *strTrace, char
*strComment);
```

## 内部BASICシンタックス

```
Function AmCreateProjectTrace(IProjectId As Long, IWorkOrderId As Long, iTraceType
As Long, IDutyId As Long, strModLinkLabel As String, strTrace As String, strComment
As String) As Long
```

## 用途

バージョン: 4.00

	使用
AssetCenter API	<input checked="" type="checkbox"/>
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	<input checked="" type="checkbox"/>
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	<input checked="" type="checkbox"/>

## パラメータ

- IProjectId: トレースの情報を取得するプロジェクトのID
- IWorkOrderId: トレースの情報を取得する作業指示のID
- iTraceType: トレースのタイプ
  - 1=接続
  - 2=切断
- IDutyId: 用途のID。作業指示内に表示されます。
- strModLinkLabel: 作業指示用に使われるコメント

- strTrace : 作業指示用に使われるトレースの説明の文字列
- strComment : 作業指示に添付されるコメント

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateReceiptFromPOOrder()

この関数は発注から受領を作成します。

## APIシンタックス

```
long AmCreateReceiptFromPOOrder(long hApiCnxBase, long IPOOrderId);
```

## 内部BASICシンタックス

```
Function AmCreateReceiptFromPOOrder(IPOOrderId As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IPOrderId : 関連する発注のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateRecord()

デフォルトの値を取る空のレコードをテーブル内に作成します。このレコードは、データベースに挿入した時に作成されます。

## APIシンタックス

```
long AmCreateRecord(long hApiCnxBase, char *strTable);
```

## 内部BASICシンタックス

```
Function AmCreateRecord(strTable As String) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strTable : レコード作成先のテーブルのSQL名

## 例

次の例ではデータベースに従業員を作成します。

```
Dim lErr As Long
Dim hRecord As Long
hRecord = amCreateRecord("amEmpIDept")
lErr = amSetFieldStrValue(hRecord, "Name", "Doe")
lErr = amSetFieldStrValue(hRecord, "FirstName", "John")
lErr = amInsertRecord(hRecord)
```

## AmCreateRequestToInvoice()

調達サイクルの全オブジェクト（依頼、発注、受領、請求）を作成します。

## APIシンタックス

```
long AmCreateRequestToInvoice(long hApiCnxBase, float fQty, long lCatRefId, double dUnitPrice, char *strCur, long lRequesterId, long lCostId, long lUserId, long lStockId);
```

## 内部BASICシンタックス

```
Function AmCreateRequestToInvoice(fQty As Single, lCatRefId As Long, dUnitPrice As Double, strCur As String, lRequesterId As Long, lCostId As Long, lUserId As Long, lStockId As Long) As Long
```

## 用途

バージョン : 4.00

使用

AssetCenter API



[リンクまたはフィールドの設定スクリプト](#)

使用

「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- fQty：発注、受領、請求する数量（パッケージ単位）
- ICatRefId：カタログリファレンスのID
- dUnitPrice：カタログリファレンスの単価
- strCur：カタログリファレンスの価格用の通貨コード
- IRequesterId：依頼者のID
- ICostId：影響を受けるコストセンタのID
- IUserId：注文品のユーザID
- IStockId：受領品の配達先在庫のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

次の関数を順番に呼び出すのと同じ操作になります。`amCreateRequestToReceipt`, `amCreateOrUpdateInvoiceFromReceipt`

## AmCreateRequestToPOrder()

調達サイクルのオブジェクト（依頼、発注）を作成します。

## APIシンタックス

```
long AmCreateRequestToPOrder(long hApiCnxBase, float fQty, long lCatRefId, double
dUnitPrice, char *strCur, long lRequesterId, long lCostId, long lUserId, long lStockId);
```

## 内部BASICシンタックス

```
Function AmCreateRequestToPOrder(fQty As Single, lCatRefId As Long, dUnitPrice As
Double, strCur As String, lRequesterId As Long, lCostId As Long, lUserId As Long,
lStockId As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- fQty：発注する数量（パッケージ単位）
- lCatRefId：カタログリファレンスのID
- dUnitPrice：カタログリファレンスの単価
- strCur：カタログリファレンスの価格用の通貨コード
- lRequesterId：依頼者のID
- lCostId：影響を受けるコストセンタのID
- lUserId：注文品のユーザID
- lStockId：受領品の配達先在庫のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateRequestToReceipt()

調達サイクルのオブジェクト（依頼、発注、受領）を作成します。

### APIシンタックス

```
long AmCreateRequestToReceipt(long hApiCnxBase, float fQty, long ICatRefId, double dUnitPrice, char *strCur, long IRequesterId, long ICostId, long IUserId, long IStockId);
```

### 内部BASICシンタックス

```
Function AmCreateRequestToReceipt(fQty As Single, ICatRefId As Long, dUnitPrice As Double, strCur As String, IRequesterId As Long, ICostId As Long, IUserId As Long, IStockId As Long) As Long
```

### 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- fQty：発注し受領する数量（パッケージ単位）
- ICatRefId：カタログリファレンスのID

- dUnitPrice : カタログリファレンスの単価
- strCur : カタログリファレンスの価格用の通貨コード
- IRequesterId : 依頼者のID
- ICostId : 影響を受けるコストセンタのID
- IUserId : 注文品のユーザID
- IStockId : 受領品の配達先在庫のID

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

次の関数を順番に呼び出すのと同じ操作になります。amCreateRequestToPOOrder, amCreateReceiptFromPOOrder

## AmCreateReturnFromReceipt()

受領伝票から返却伝票を作成します。

## APIシンタックス

```
long AmCreateReturnFromReceipt(long hApiCnxBase, long IRecptId);
```

## 内部BASICシンタックス

```
Function AmCreateReturnFromReceipt(IRecptId As Long) As Long
```

## 用途

バージョン : 4.00

## 使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IRecptId : 受領のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCreateTraceHist()

AmCreateTraceHist APIは、ソースデバイス/ケーブルからターゲットデバイス/ケーブルへの既存の接続に基づいて、トレース履歴とトレースの処理を作成します。

## APIシンタックス

```
long AmCreateTraceHist(long hApiCnxBase, long ISrcLinkId, long IDestLinkId, long iTraceDir, long ICabTraceOutId, char *strComment);
```

## 内部BASICシンタックス

```
Function AmCreateTraceHist(ISrcLinkId As Long, IDestLinkId As Long, iTraceDir As Long, ICabTraceOutId As Long, strComment As String) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- ISrcLinkId：ソースケーブルリンクに割り当てられているケーブルリンクのID
- IDestLinkId：ターゲットケーブルリンクに割り当てられているケーブルリンクのID
- iTraceDir：接続の方向
  - 0=ホストからユーザへの方向
  - 1=ユーザからホストへの方向
- ICabTraceOutId：ケーブルトレースの説明のID
- strComment：トレースの処理に関連するコメント

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmCryptPassword()

ログインとパスワードで識別されるユーザのパスワードを暗号化します。

## APIシンタックス

```
long AmCryptPassword(char *strUser, char *strPasswd, char *pStrCrypted, long lpStrCrypted);
```

## 内部BASICシンタックス

Function AmCryptPassword(strUser As String, strPasswd As String) As String

### 用途

バージョン : 3.5

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- strUser : 暗号化するパスワードを使用するユーザのログイン
- strPasswd : 暗号化するパスワード (普通のテキストで)

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数 (必要に応じて[AmLastErrorMsg\(\)](#)関数も) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCurrentDate()

クライアントのコンピュータの現在の日付を返します。

## APIシンタックス

```
long AmCurrentDate();
```

## 内部BASICシンタックス

Function AmCurrentDate() As Date

### 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 注

タイムゾーンを使用するようにデータベースが設定されている場合、この関数の動作は、直接AssetCenterで呼び出されたか、または外部プログラムにより呼び出されたかにより変化します。AssetCenterでは、この関数はBasic Now()と同様に動作します。外部プログラムからの場合、この関数が返す値はGMT+0のタイムゾーンで表記され、夏時間との時間差は考慮されません。

## AmCurrentIsoLang()

AssetCenter内で使用されている言語のISO言語コードを返します（英語は「en」、フランス語は「fr」、など）。

## APIシンタックス

```
long AmCurrentIsoLang(char *pstrLanguage, long ILanguage);
```

## 内部BASICシンタックス

```
Function AmCurrentIsoLang() As String
```

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCurrentLanguage()

AssetCenterの言語バージョンを返します（英語バージョンの場合は「US」、フランス語バージョンの場合は「FRな」など）。

## APIシンタックス

```
long AmCurrentLanguage(char *pstrLanguage, long ILanguage);
```

## 内部BASICシンタックス

Function AmCurrentLanguage() As String

### 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmCurrentServerDate()

サーバの現在の日付を返します。

### APIシンタックス

long AmCurrentServerDate(long hApiCnxBase);

## 内部BASICシンタックス

Function AmCurrentServerDate() As Date

## 用途

バージョン : 3.5

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDateAdd()

開始日に実際の経過時間を追加して、新たな日付を計算します。

## APIシンタックス

```
long AmDateAdd(long tmStart, long tsDuration);
```

## 内部BASICシンタックス

```
Function AmDateAdd(tmStart As Date, tsDuration As Long) As Date
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- tmStart : 経過時間を追加する日付
- tsDuration : tmStartの日付に追加する時間 (秒単位)

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数 (必要に応じてAmLastErrorMsg()関数も) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

以下の例は、関数amDateAdd() と関数amDateAddLogical()の違いを示しています。各関数は30日の期間を日付1/1/1999 (1999年1月1日) に追加します。

AmDateAddは実際の期間 (この例では30日) を追加します。

```
RetVal=AmDateAdd("1999/01/01", 2592000)
```

関数は次の値を返します。

```
1999/01/31
```

AmDateAddLogicalは論理期間を追加します。この例では30日 (=1ヶ月) です。

```
RetVal=AmDateAddLogical("1999/01/01", 2592000)
```

関数は次の値を返します。

```
1999/02/01
```

## AmDateAddLogical()

開始日に論理上の経過時間を追加して、新たな日付を計算します（1ヶ月を30日として計算します）。

### APIシンタックス

```
long AmDateAddLogical(long tmStart, long tsDuration);
```

### 内部BASICシンタックス

```
Function AmDateAddLogical(tmStart As Date, tsDuration As Long) As Date
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- tmStart：経過時間を追加する日付
- tsDuration：tmStartの日付に追加する時間（秒単位）

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

以下の例は、関数amDateAdd() と関数amDateAddLogical()の違いを示しています。各関数は30日の期間を日付1/1/1999（1999年1月1日）に追加します。

AmDateAddは実際の期間（この例では30日）を追加します。

```
RetVal=AmDateAdd("1999/01/01", 2592000)
```

関数は次の値を返します。

```
1999/01/31
```

AmDateAddLogicalは論理期間を追加します。この例では30日（=1ヶ月）です。

```
RetVal=AmDateAddLogical("1999/01/01", 2592000)
```

関数は次の値を返します。

```
1999/02/01
```

## AmDateDiff()

指定した2つの日付の間の時間を秒単位で計算します。

### APIシンタックス

```
long AmDateDiff(long tmEnd, long tmStart);
```

### 内部BASICシンタックス

```
Function AmDateDiff(tmEnd As Date, tmStart As Date) As Date
```

### 用途

バージョン：3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- tmEnd : 計算する期間の終了日
- tmStart : 計算する期間の開始日

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

1998年1月1日から1999年1月1日までの間に経過する時間を計算します。

```
AmDateDiff("1998/01/01 00:00:00", "1999/01/01 00:00:00")
```

## AmDbGetDate()

AQLクエリの結果を日付形式で返します。クエリが結果を返さない場合は「0」が返され、エラーは発生しません。

## APIシンタックス

```
long AmDbGetDate(long hApiCnxBase, char *strQuery);
```

## 内部BASICシンタックス

Function AmDbGetDate(strQuery As String) As Date

### 用途

バージョン : 3.5

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- strQuery : 実行するAQLクエリの全文

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDbGetDouble()

AQLクエリの結果を倍精度の数値で返します。クエリが結果を返さない場合は「0」が返され、エラーは発生しません。

### APIシンタックス

```
double AmDbGetDouble(long hApiCnxBase, char *strQuery);
```

## 内部BASICシンタックス

Function AmDbGetDouble(strQuery As String) As Double

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strQuery : 実行するAQLクエリの全文

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDbGetList()

AQLクエリの結果をリストとして返します。AQLクエリで選択できる要素数は最高で99個です。

## APIシンタックス

```
long AmDbGetList(long hApiCnxBase, char *strQuery, char *pstrResult, long lResult,
char *strColSep, char *strLineSep, char *strIdSep);
```

## 内部BASICシンタックス

```
Function AmDbGetList(strQuery As String, strColSep As String, strLineSep As String,
strIdSep As String) As String
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strQuery : 実行するAQLクエリ
- strColSep : クエリの結果の列区切りに使用する文字
- strLineSep : クエリの結果の行区切りに使用する文字
- strIdSep : クエリの結果のID区切りに使用する文字

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDbGetListEx()

AQLクエリの結果をリストとして返します。AmDbGetList関数と異なり、AQLクエリで選択できる要素数に制限がありません。

### APIシンタックス

```
long AmDbGetListEx(long hApiCnxBase, char *strQuery, char *pstrResult, long IResult, char *strColSep, char *strLineSep, char *strIdSep);
```

### 内部BASICシンタックス

```
Function AmDbGetListEx(strQuery As String, strColSep As String, strLineSep As String, strIdSep As String) As String
```

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strQuery : 実行するAQLクエリ
- strColSep : クエリの結果の列区切りに使用する文字
- strLineSep : クエリの結果の行区切りに使用する文字
- strIdSep : クエリの結果のID区切りに使用する文字

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDbGetLong()

AQLクエリの結果を返します。クエリが結果を返さない場合は「0」が返され、エラーは発生しません。

## APIシンタックス

```
long AmDbGetLong(long hApiCnxBase, char *strQuery);
```

## 内部BASICシンタックス

```
Function AmDbGetLong(strQuery As String) As Long
```

## 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strQuery：実行するAQLクエリの全文

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

製品のサプライヤのIDを返します。

```
AmDbGetLong("SELECT ISupId FROM amProdSupp WHERE IProdId="+Str([ProdId])+")
```

## AmDbGetPk()

AQLクエリのWHERE句に従って、テーブルの主キーを返します。クエリが結果を返さない場合は「0」が返され、エラーは発生しません。

## APIシンタックス

```
long AmDbGetPk(long hApiCnxBase, char *strTableName, char *strWhere);
```

## 内部BASICシンタックス

```
Function AmDbGetPk(strTableName As String, strWhere As String) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	

	使用
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strTableName : 主キーを取得するテーブルのSQL名
- strWhere : AQLクエリのWHERE句

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDbGetString()

AQLクエリの結果を書式化した文字列として返します。AQLクエリで選択できる要素数は最高で99個です。

### ⚠ 警告:

文字列タイプの1フィールド値を取得するためにはこの関数を使用しないでください。この関数は、AmDbGetListまたはAmDbGetListExに類似しています。

## APIシンタックス

```
long AmDbGetString(long hApiCnxBase, char *strQuery, char *pstrResult, long lResult, char *strColSep, char *strLineSep);
```

## 内部BASICシンタックス

Function AmDbGetString(strQuery As String, strColSep As String, strLineSep As String)  
As String

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strQuery：実行するAQLクエリ
- strColSep：クエリで取得した文字列の列区切りに使用する文字
- strLineSep：クエリで取得した文字列の行区切りに使用する文字

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 注

APIシンタックスでは、パラメータIResultは関数の結果値の予想されるサイズを含みません。

## 例

```
Dim strList As String
strList = amDbGetList("Select Name, FullName from amEmplDept Where Name Like 'C%'", "|", ",", "=")
```

これは次の文字列を返します。

```
Carpenter|Taltek/I.S. Department/Carpenter/, Jerome/, DEMO-M016/=23459,Chavez|
Taltek/I.S. Department/Chavez/, Philip/, DEMO-M014/=23460,Chouraqui|Taltek/Sales
/Los Angeles Agency/Chouraqui/, Thomas/, DEMO-M017/=23491,Cipriani|Taltek/Sal
es/Los Angeles Agency/Cipriani/, Fred/, DEMO-M018/=23492,Clech|Taltek/Sales/Bur
bank Agency/Clech/, Richard/, DEMO-M021/=23482,Colombo|Taltek/Finance/Colom
bo/, Gerald/, DEMO-M022/=23441
```

コンマの前には、エスケープ文字「\」を使用します。

amDbGetString()関数で実行する同じクエリは、エスケープ文字を追加しないので、リストを埋めるには不適切になります。例えば、

```
amDbGetString("Select FullName from amEmplDept Where Name Like 'C%'", "|", chr(10), "")
```

は以下の結果を表示します。

```
/Taltek/I.S. Department/Carpenter, Jerome, DEMO-M016/
/Taltek/I.S. Department/Chavez, Philip, DEMO-M014/
/Taltek/Sales/Los Angeles Agency/Chouraqui, Thomas, DEMO-M017/
/Taltek/Sales/Los Angeles Agency/Cipriani, Fred, DEMO-M018/
/Taltek/Sales/Burbank Agency/Clech, Richard, DEMO-M021/
/Taltek/Finance/Colombo, Gerald, DEMO-M022/
```

## AmDbGetStringEx()

AQLクエリの結果を文字列として返します。AmDbGetString関数と異なり、AQLクエリで選択できる要素数に制限がありません。

### 警告:

文字列タイプの1フィールド値を取得するためにはこの関数を使用しないでください。この関数は、AmDbGetListまたはAmDbGetListExに類似しています。

## APIシンタックス

```
long AmDbGetStringEx(long hApiCnxBase, char *strQuery, char *pstrResult, long lResult,
char *strColSep, char *strLineSep);
```

## 内部BASICシンタックス

```
Function AmDbGetStringEx(strQuery As String, strColSep As String, strLineSep As
String) As String
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strQuery : 実行するAQLクエリ
- strColSep : クエリで取得した文字列の列区切りに使用する文字
- strLineSep : クエリで取得した文字列の行区切りに使用する文字

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDeadline()

カレンダー、開始日、および作業時間の秒数から期限を出算します。

### APIシンタックス

```
long AmDeadline(char *strCalendarSqlName, long tmStart, long tsDuration);
```

### 内部BASICシンタックス

```
Function AmDeadline(strCalendarSqlName As String, tmStart As Date, tsDuration As Long) As Date
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strCalendarSqlName : 期限の計算に使う業務用カレンダーのSQL名
- tmStart : 作業期間の開始日
- tsDuration : 開始日以降の作業時間の秒数

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

「Calendar\_Paris」というSQL名のカレンダーに基づいて期限を計算します。期間の開始日は1998年9月1日午前8時、秒数は450,000です。

```
AmDeadLine("Calendar_Paris", "1998/09/01 08:00:00", 450000)
```

期限は、1998年9月22日午後6時になります。

## AmDecrementLogLevel()

ウィザードの最終ページのログウィンドウの階層内で1つ上のレベルに移動します。

## 内部BASICシンタックス

Function AmDecrementLogLevel() As Long

## 用途

バージョン : 3.5

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト 

---

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDefAssignee()

指定した従業員グループのデフォルトのチケット責任者のID番号を検索します。

### APIシンタックス

```
long AmDefAssignee(long hApiCnxBase, long IGroupId);
```

### 内部BASICシンタックス

```
Function AmDefAssignee(IGroupId As Long) As Long
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IGroupId：従業員グループのID番号

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

従業員グループのデフォルトの責任者のIDを返します。

```
AmDefAssignee([IGroupId])
```

次の例のように、IDの数値を直接入力することもできます。

```
AmDefAssignee(24)
```

## AmDefaultCurrency()

AssetCenterで使用しているデフォルトの通貨を返します。

## APIシンタックス

```
long AmDefaultCurrency(long hApiCnxBase, char *return, long lreturn);
```

## 内部BASICシンタックス

```
Function AmDefaultCurrency() As String
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	

## 使用

---

 ウィザードの「FINISH.DO」スクリプト
 

---



## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmDefEscalationScheme()

場所とヘルプデスクチケットの優先度に応じて、デフォルトのエスカレーション処理を検索します。

## APIシンタックス

```
long AmDefEscalationScheme(long hApiCnxBase, char *strLocFullName, long ISeverityLvl);
```

## 内部BASICシンタックス

```
Function AmDefEscalationScheme(strLocFullName As String, ISeverityLvl As Long) As Long
```

## 用途

バージョン：3.00

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	

---

## ウィザードの「FINISH.DO」スクリプト

---



### パラメータ

- strLocFullName : 場所の完全名
- ISeverityLvl : 優先度の値

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

場所と優先度に応じて、デフォルトのエスカレーション処理のIDを返します。

```
AmDefEscalationScheme([Asset.Location.FullName], [Severity.ISeverityLvl])
```

次の例のように、パラメータの値を直接入力することもできます。

```
AmDefEscalationScheme ("/Location/", 24)
```

## AmDefGroup()

問題のタイプ、場所、およびメンテナンス契約に応じて、デフォルトのヘルプデスクグループのID番号を返します。

### APIシンタックス

```
long AmDefGroup(long hApiCnxBase, long IProblemClassId, char *strLocFullName, long IAssetMainCntId);
```

## 内部BASICシンタックス

Function AmDefGroup(IProblemClassId As Long, strLocFullName As String,  
IAssetMainCntId As Long) As Long

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IProblemClassId : 問題のタイプのID番号
- strLocFullName : 場所の完全名
- IAssetMainCntId : メンテナンス契約のID番号

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

デフォルトのヘルプデスクグループを定義する方法は以下の通りです。

- 1 関数は、ヘルプデスクチケットの問題のタイプに関連するヘルプデスクグループを検索します。

- 2 検索されたグループの中から、関数は資産の場所に最も「近い」場所に関連付けられているヘルプデスクグループを検索します。資産の場所にヘルプデスクグループがない場合は、資産の場所の親所在地で探し、最終的にはルート場所まで検索します。
- 3 上記の方法でヘルプデスクグループが見つからないと、DBMSが2重の外結合をサポートする場合は、関数は場所に関連付けられていないグループを検索します。

2重の外結合をサポートするDBMSの詳細は、マニュアル『ヘルプデスク』の、「参考情報（ヘルプデスク）」の章、「外部二重結合をサポートするDBMS」の節を参照してください。

- 4 DBMSが2重の外結合をサポートする場合、関数は最初に検索されたグループの中から、資産のメンテナンス契約に関連付けられているヘルプデスクグループを検索します。
- 5 グループが見つからない場合、関数は階層構造で1段階高い問題のタイプで手順1、2、3、4を繰り返し、グループが見つかるまでルートの問題のタイプまで検索します。

## 例

問題のタイプ、場所、およびメンテナンス契約の3つのパラメータに応じて、デフォルトのヘルプデスクグループのID番号を返します。

```
AmDefGroup([ProblemClass.IPbClassId],[Asset.Location.FullName],[Asset.IMaintCnt
rId])
```

次の例のように、パラメータのID番号の数値を直接入力することもできます。

```
AmDefGroup(0, [Asset.Location.FullName], 0)
```

## AmDeleteLink()

レコードのリンクを削除します。

## APIシンタックス

```
long AmDeleteLink(long hApiRecord, char *strLinkName, long hApiRecDest);
```

## 内部BASICシンタックス

Function AmDeleteLink(hApiRecord As Long, strLinkName As String, hApiRecDest As Long) As Long

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiRecord : 削除するリンクを含むレコードのハンドル
- strLinkName : 削除するリンクのSQL名
- hApiRecDest : リンクのターゲットレコードのハンドル

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmDeleteRecord()

データベースから1つのレコードを削除します。

## APIシンタックス

long AmDeleteRecord(long hApiRecord);

## 内部BASICシンタックス

Function AmDeleteRecord(hApiRecord As Long) As Long

### 用途

バージョン：2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiRecord：削除するレコードのハンドル

### 戻りコード

- 0：成功
- 0以外：エラーコード

## AmDisconnectTrace()

AmDisconnectTrace APIはユーザノード (IEndId) とホストノード (IStartId) 間のトレースを切断します。ノードがトレースの最後にある場合、ノードはケーブルリンクテーブルから削除されます。切断後、トレース履歴とトレースの処理を作成します。

### APIシンタックス

```
long AmDisconnectTrace(long hApiCnxBase, long IStartId, long IEndId, char
*strComment, long ICabTraceOutId);
```

## 内部BASICシンタックス

Function AmDisconnectTrace(IStartId As Long, IEndId As Long, strComment As String, ICabTraceOutId As Long) As Long

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IStartId：切断されるホストの接続ID
- IEndId：切断されるユーザの接続ID
- strComment：新規接続と切断を示すトレース処理の文字列
- ICabTraceOutId：ケーブルのトレースの説明のID

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmDuplicateRecord()

この関数はレコードを複製します。

## APIシンタックス

long AmDuplicateRecord(long hApiRecord, long bInsert);

## 内部BASICシンタックス

Function AmDuplicateRecord(hApiRecord As Long, bInsert As Long) As Long

### 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiRecord：複製するレコードのハンドル
- bInsert：このパラメータでは、複製されたレコードを即時に挿入するか (=1) しないか (=0) を指定します。

### 戻りコード

- 0：成功
- 0以外：エラーコード

## AmEndOfNthBusinessDay()

カレンダーに応じて、指定した日付からn番目の日付（IDayCountで特定）の最後の業務時間を割り出します。

### APIシンタックス

long AmEndOfNthBusinessDay(char \*strCalendarSqlName, long tmStart, long lDayCount);

## 内部BASICシンタックス

Function AmEndOfNthBusinessDay(strCalendarSqlName As String, tmStart As Date, lDayCount As Long) As Date

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strCalendarSqlName : 計算に使用するカレンダーの名前
- tmStart : 計算の開始日
- lDayCount : 計算する時にdStartに追加する業務日数

### 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmEnumValList()

システムリストデータのすべての値の文字列を返します。各値は、アルファベット順に並べ替えられ、strLineSepパラメータの文字で区切られます。

リストデータの値の中に区切り文字またはバックスラッシュ ( / ) が含まれている場合は、バックスラッシュが接頭文字として使われます。

## APIシンタックス

```
long AmEnumValList(long hApiCnxBase, char *strEnumName, char *pstrValList, long lValList, long bNoCase, char *strLineSep);
```

## 内部BASICシンタックス

```
Function AmEnumValList(strEnumName As String, bNoCase As Long, strLineSep As String) As String
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strEnumName：値を取得するシステムリストデータのSQL名
- bNoCase：アルファベット順の並べ替えが大文字小文字を区別するか (=1) しないか (=0) を指定します。
- strLineSep：リストデータの各値を区切る文字

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmEvalScript()

現在のコンテキストでスクリプトの名前からスクリプトを評価します。この関数には2つの用途があります。

- システムスクリプトを評価する（デフォルト値、必須、など）。
- スクリプトライブラリから関数を呼び出す。

## 内部BASICシンタックス

Function AmEvalScript(strScriptName As String, strObject As String, strPath As String, ...) As Variant

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strScriptName : 評価するスクリプト名。1番目の用途では、システムスクリプト名（DefVal、など）を入力します。2番目の用途では、スクリプトライブラリの一部を成す関数の名前（ライブラリの名前はパラメータstrObjectで指定）を入力します。
- strObject : スクリプトに関連するオブジェクト。フィールドのSQL名またはライブラリ名です。

- `strPath` : このオプションパラメータは、スクリプト評価のコンテキストを移動させるためのパス (リンク.リンク.リンク...) を指定します。このオプションは、関数を2番目の用途に使用する場合には機能しません。
- ... : スクリプトライブラリから関数を呼び出す際、呼び出された関数にパラメータを渡します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数 (必要に応じて`AmLastErrorMsg()`関数も) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

使用可能なシステムスクリプト名のリストは次の通りです。

- テーブル用 : `IsValidScript`、`IsRelevantScript`
- フィールド用 : `DefValScript`、`MandatoryScript`、`HistorizedScript`、`ReadOnlyScript`、`IrrelevantScript`
- リンク用 : `HistorizedScript`、`FilterScript`、`IrrelevantScript`
- 任意管理項目用 : `DefValScript`、`MandatoryScript`、`AvailableScript`、`HistorizedScript`

## AmExecTransition()

現在のページから有効なトランジションを起動します。

## 内部BASICシンタックス

```
Function AmExecTransition(strTransName As String) As Long
```

## 用途

バージョン : 3.00

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strTransName : ウィザードのスクリプトに定義されているトランジションの名前

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmExecuteActionById()

IDでアクションを識別し、そのアクションを実行します。

## APIシンタックス

```
long AmExecuteActionById(long IActionId, char *strTableName, long IRecordId);
```

## 内部BASICシンタックス

```
Function AmExecuteActionById(IActionId As Long, strTableName As String, IRecordId As Long) As Long
```

## 用途

バージョン : 3.00

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IActionId : 実行するアクションのID
- strTableName : コンテキストを指定するアクションの場合は、アクションを実行するテーブルのSQL名。コンテキストを指定するアクションでこのパラメータを省略すると、この関数は機能しません。コンテキストを指定しないアクションの場合は、このパラメータは解釈されないため、必要ありません。
- IRecordId : アクションで使用する可能性のあるレコードのID。コンテキストを指定しないアクションの場合は、このパラメータは解釈されないため、必要ありません。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmExecuteActionByName()

SQL名でアクションを識別し、そのアクションを実行します。

## APIシンタックス

```
long AmExecuteActionByName(char *strSqlName, char *strTableName, long IRecordId);
```

## 内部BASICシンタックス

```
Function AmExecuteActionByName(strSqlName As String, strTableName As String, IRecordId As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strSqlName : 実行するアクションのSQL名
- strTableName : コンテキストを指定するアクションの場合は、アクションを実行するテーブルのSQL名。コンテキストを指定するアクションでこのパラメータを省略すると、この関数は機能しません。コンテキストを指定しないアクションの場合は、このパラメータは解釈されないため、必要ありません。
- IRecordId : アクションで使用する可能性のあるレコードのID。コンテキストを指定しないアクションの場合は、このパラメータは解釈されないため、必要ありません。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmExportDocument()

レコードに添付されている書類を出力します。

## APIシンタックス

```
long AmExportDocument(long hApiCnxBase, long IDocId, char *strFileName);
```

## 内部BASICシンタックス

Function AmExportDocument(IDocId As Long, strFileName As String) As Long

### 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- IDocId：出力する書類のID
- strFileName：出力する書類の名前。書類のテーブルのFileNameフィールド内の名前を使用します。

### 戻りコード

- 0：成功
- 0以外：エラーコード

## AmFindCable()

AmFindCable APIは、あるユーザ（IUserId）とホスト（IHostId）の場所間にある使用可能な次のケーブルを検索します。ケーブルのタイプ（strCabType）と役割（strCableRole）は、特定のタイプと役割でなければなりません。またケーブルのステータスは「使用可能」でなければなりません。ケーブルはIDで昇順に並べ替えられ、前のケーブルのID（IPrevCabId）よりもIDの大きいケーブルのみが選択されます。

## APIシンタックス

```
long AmFindCable(long hApiCnxBase, long IPrevCableId, char *strCabType, long IUserId,
long IHostId, char *strCableRole);
```

## 内部BASICシンタックス

```
Function AmFindCable(IPrevCableId As Long, strCabType As String, IUserId As Long,
IHostId As Long, strCableRole As String) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IPrevCableId：前のケーブルのID
- strCabType：検索するケーブルのタイプ
- IUserId：ユーザの場所のID
- IHostId：ホストの場所のID
- strCableRole：検索するケーブルの役割

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmFindDevice()

AmFindDevice APIは、あるタイプ ( strDevType ) で一定の場所 ( ILocationId ) にあるデバイスを検索します。デバイスはIDで昇順に並べ替えられ、前のデバイスID ( IPrevDeviceId ) よりもIDの大きいデバイスが選択されます。

### APIシンタックス

```
long AmFindDevice(long hApiCnxBase, long IPrevDeviceId, char *strDeviceType, long ILocationId);
```

### 内部BASICシンタックス

```
Function AmFindDevice(IPrevDeviceId As Long, strDeviceType As String, ILocationId As Long) As Long
```

### 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IPrevDeviceId : 検索される旧デバイスのID。検索を開始するには、値「0」が使用されます。
- strDeviceType : 検索するデバイスのタイプ
- ILocationId : 検索する場所のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmFindRootLink()

トレースのルートリンクを取得します。

## APIシンタックス

```
long AmFindRootLink(long hApiCnxBase, long ILinkId);
```

## 内部BASICシンタックス

```
Function AmFindRootLink(ILinkId As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- ILinkId：演算に関連するリンクのID

## 戻りコード

ルート（ケーブル）リンクのIDを返します。

## AmFindTermDevice()

AmFindTermDevice APIは、ある成端場所（ITermField）で一定の役割（strCableRole）を果たす使用可能なデバイスを検索します。デバイスは連続番号で昇順に並び替えられ、旧連続番号（strPrevTermSeq）よりも番号の大きい資産のみが選択されます。ピン型のデバイス（bPinBased=1）では、必要なピンの総数（iPinPortCount）とデバイス上にあるピンの残り総数を比較します。ポート型のデバイス（bPinBased=0）では、デバイス上に最低1つのポートがあり、このポートのホスト側またはユーザ側が使用可能であるかどうかをフラグで示します（bCheckAvail = 0 - user device, bCheckAvail = 1 - host device）

## APIシンタックス

```
long AmFindTermDevice(long hApiCnxBase, long iPrevTermSeq, long ITermFieldId,
char *strCableRole, long bPinBased, long iPinPortCount, long bCheckAvail);
```

## 内部BASICシンタックス

```
Function AmFindTermDevice(iPrevTermSeq As Long, ITermFieldId As Long,
strCableRole As String, bPinBased As Long, iPinPortCount As Long, bCheckAvail As
Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- iPrevTermSeq : 検索される成端場所の前の手順。「0」は検索を開始します。
- lTermFieldId : 成端場所のID
- strCableRole : 検索するケーブルの役割
- bPinBased : デバイスがピン型またはポート型であるかを指定します。
- iPinPortCount : ピン型のデバイスでは、このパラメータは仮想ポート作成のために必要なピンの総数です。ポート型のデバイスでは必要な各ポート用にAPIが呼び出されるため、このパラメータは「1」です。
- bCheckAvail : このパラメータでは、ポートのどの側面が使用可能であるべきかを指定できます。
  - 0=ユーザデバイス、使用可能なホスト側を検索します。
  - 1=ホストデバイス、使用可能なユーザ側を検索します。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmFindTermField()

AmFindTermField APIはある用途 (IDutyId) をある場所 (ILocationId) で果たす成端場所を検索します。IPrevTermFieldIdの値が「0」よりも大きい場合、このAPIはある一定の用途と場所の追加成端場所の検索を続行します。

## APIシンタックス

```
long AmFindTermField(long hApiCnxBase, long lDutyId, long lLocationId, long lPrevTermFieldId);
```

## 内部BASICシンタックス

Function AmFindTermField(IDutyId As Long, ILocationId As Long, IPrevTermFieldId As Long) As Long

### 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IDutyId：このパラメータは検索する機能を定義します。
- ILocationId：検索する場所のID
- IPrevTermFieldId：成端場所のID

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGenSqlName()

普通の文字列からSQL名を作成します。スペース（空白文字）はアンダースコア（\_）に置き換えられます。この関数は、特に任意管理項目の名前からデフォルトのSQL名を作成する場合に便利です。

## APIシンタックス

```
long AmGenSqlName(char *return, long lreturn, char *strText);
```

## 内部BASICシンタックス

```
Function AmGenSqlName(strText As String) As String
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strText : SQL名の作成に使う文字列

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

AssetCenterデータベース内の「Label」というオブジェクトのデフォルトのSQL名を定義します。

```
RetVal=AmGenSQLName([Label])
```

## AmGetCatRef()

この関数は、あるモデル用の有効なカタログ外リファレンスを検索します（有効期限が厳守されます）。次の規則があります。

- ```
CatProduct.IModelId=IModelId
```
- ```
CatProduct.IParentId=0
```

関数は、仮作成されていないリファレンスを優先して返します。リファレンスが見つからず、パラメータbCreateの値が「1」であると、新規にカタログ外リファレンスと、製品（モデルを参照する）が作成されます。

## APIシンタックス

```
long AmGetCatRef(long hApiCnxBase, long IModelId, long bCreate);
```

## 内部BASICシンタックス

```
Function AmGetCatRef(IModelId As Long, bCreate As Long) As Long
```

## 用途

バージョン：4.1.0

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IModelId：演算に関連するモデルのID

- bCreate : 検索しても結果が返されない場合に、このパラメータはカタログ外のリファレンスが作成されるかどうかを指定します。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

---

 **注意:**

関数にサプライヤを指定する必要はありません。検索は、全サプライヤのカタログ外のリファレンスを対象に実行されます。

---

## AmGetCatRefFromCatProduct()

この関数は、関数amGetCatRefと同じですが、相違点は特定の製品に対して検索が実行される点です。

## APIシンタックス

```
long AmGetCatRefFromCatProduct(long hApiCnxBase, long lCatProductId, long bCreate);
```

## 内部BASICシンタックス

```
Function AmGetCatRefFromCatProduct(lCatProductId As Long, bCreate As Long) As Long
```

## 用途

バージョン : 4.1.0

## 使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- ICatProductId : 演算に関連する製品のID
- bCreate : 検索しても結果が返されない場合に、このパラメータはカタログ外のリファレンスが作成されるかどうかを指定します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetComputeString()

指定したレコードの説明文字列をテンプレートに従って返します。

## APIシンタックス

```
long AmGetComputeString(long hApiCnxBase, char *strTableName, long IRecordId,
char *strTemplate, char *pstrComputeString, long IComputeString);
```

## 内部BASICシンタックス

```
Function AmGetComputeString(strTableName As String, IRecordId As Long, strTemplate
As String) As String
```

## 用途

バージョン：3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strTableName：説明文字列を取得するレコードのテーブルのSQL名
- IRecordId：テーブル内のレコードのID
- strTemplate：説明文字列に使うテンプレート（文字列形式）

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
RetVal = amGetComputeString("amEmplDept", [IEmplDeptId], "[Name], [FirstName]"
)
```

## AmGetCurrentNTDomain()

現在のログインのNTドメイン名を返します。

## APIシンタックス

```
long AmGetCurrentNTDomain(char *pstrDomain, long lDomain);
```

## 内部BASICシンタックス

```
Function AmGetCurrentNTDomain() As String
```

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
RetVal = amGetCurrentNTDomain()
```

## AmGetCurrentNTUser()

Windows（NTまたは2000）に接続しているユーザのログインを取得します。

## APIシンタックス

```
long AmGetCurrentNTUser(char *pstrUser, long lUser);
```

## 内部BASICシンタックス

```
Function AmGetCurrentNTUser() As String
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFeat()

名前に基づいて任意管理項目オブジェクトを作成し、そのオブジェクトのハンドルを返します。

## APIシンタックス

```
long AmGetFeat(long hApiTable, long lPos);
```

## 内部BASICシンタックス

Function AmGetFeat(hApiTable As Long, IPos As Long) As Long

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiTable : テーブルのハンドル
- IPos : テーブル内の任意管理項目の位置

## AmGetFeatCount()

hApiTableパラメータで指定するテーブルの任意管理項目数を返します。

## APIシンタックス

long AmGetFeatCount(long hApiTable);

## 内部BASICシンタックス

Function AmGetFeatCount(hApiTable As Long) As Long

### 用途

バージョン : 3.5

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiTable : テーブルのハンドル

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetField()

クエリ、レコード、またはテーブルのハンドルからフィールドオブジェクトを作成し、そのオブジェクトのハンドルを返します。

## APIシンタックス

```
long AmGetField(long hApiObject, long IPos);
```

## 内部BASICシンタックス

```
Function AmGetField(hApiObject As Long, IPos As Long) As Long
```

## 用途

バージョン：2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiObject：クエリ、レコード、またはテーブルのハンドル
- IPos：オブジェクト内のフィールドの位置（インデックス）

## AmGetFieldCount()

現在のオブジェクトに含まれているフィールド数を返します。

## APIシンタックス

```
long AmGetFieldCount(long hApiObject);
```

## 内部BASICシンタックス

```
Function AmGetFieldCount(hApiObject As Long) As Long
```

## 用途

バージョン：2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	

	使用
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiObject : 有効なレコード、クエリ、またはテーブルのハンドル

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldValue()

現在のオブジェクトに含まれているフィールドの値を返します。値は日付（「Date」）形式で返されます（外部ツールからの場合は「Long」形式です）。

## APIシンタックス

```
long AmGetFieldValue(long hApiObject, long IFieldPos);
```

## 内部BASICシンタックス

```
Function AmGetFieldValue(hApiObject As Long, IFieldPos As Long) As Date
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiObject : クエリまたはレコードのハンドル
- IFieldPos : 現在のオブジェクトに含まれているフィールドの番号

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldDescription()

ハンドルでフィールドを識別し、そのフィールドの長い説明を、文字列（String形式）で返します。

## APIシンタックス

```
long AmGetFieldDescription(long hApiField, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

```
Function AmGetFieldDescription(hApiField As Long) As String
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 長い説明を取得するフィールドの有効なハンドル

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldDoubleValue()

現在のオブジェクトに含まれているフィールドの値を返します。値は倍精度（「Double」）形式で返されます。

## APIシンタックス

```
double AmGetFieldDoubleValue(long hApiObject, long IFieldPos);
```

## 内部BASICシンタックス

```
Function AmGetFieldDoubleValue(hApiObject As Long, IFieldPos As Long) As Double
```

## 用途

バージョン：3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiObject：クエリまたはレコードのハンドル
- IFieldPos：現在のオブジェクトに含まれているフィールドの番号

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldFormat()

この関数は、フィールドの入力タイプが次のような場合に有効です。

- システムリストデータ
- リストデータ
- 経過時間
- テーブルまたはフィールドのSQL名

次のように、各入力タイプのフォーマットを返します。

UserType	関数が返すフォーマット
システムリストデータ	リストデータ項目のリスト

UserType	関数が返すフォーマット
リストデータ	フィールドに関連付けられているリスト データの名前
経過時間	表示フォーマット
テーブルまたはフィールドのSQL名	テーブルのSQL名を保存するフィールドの SQL名

## APIシンタックス

```
long AmGetFieldFormat(long hApiField, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

```
Function AmGetFieldFormat(hApiField As Long) As String
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 「UserType」を取得するフィールドの有効なハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldFormatFromName()

フィールドの名前からフィールドの入力タイプを取得します。

### APIシンタックス

```
long AmGetFieldFormatFromName(long hApiCnxBase, char *strTableName, char *strFieldName, char *pFieldFormat, long lpFieldFormat);
```

### 内部BASICシンタックス

```
Function AmGetFieldFormatFromName(strTableName As String, strFieldName As String) As String
```

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strTableName : 処理するフィールドを含んでいるテーブルのSQL名
- strFieldName : フィールドのSQL名

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldFromName()

名前からフィールドオブジェクトを作成し、作成したフィールドオブジェクトのハンドルを返します。

## APIシンタックス

```
long AmGetFieldFromName(long hApiObject, char *strName);
```

## 内部BASICシンタックス

```
Function AmGetFieldFromName(hApiObject As Long, strName As String) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiObject : クエリ、レコード、またはテーブルのハンドル

- strName : フィールド名

## AmGetFieldLabel()

ハンドルでフィールドを識別し、そのフィールドのラベルを文字列 (String形式) で返します。

## APIシンタックス

```
long AmGetFieldLabel(long hApiField, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

```
Function AmGetFieldLabel(hApiField As Long) As String
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField : 取得するラベルの付いたフィールドの有効なハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldLabelFromName()

フィールドのSQL名からフィールドのラベルを取得します。

### APIシンタックス

```
long AmGetFieldLabelFromName(long hApiCnxBase, char *strTableName, char *strFieldName, char *pFieldLabel, long lpFieldLabel);
```

### 内部BASICシンタックス

```
Function AmGetFieldLabelFromName(strTableName As String, strFieldName As String) As String
```

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strTableName : 処理するフィールドを含んでいるテーブルのSQL名
- strFieldName : フィールドのSQL名

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldLongValue()

現在のオブジェクトに含まれているフィールドの値を返します。

## APIシンタックス

```
long AmGetFieldLongValue(long hApiObject, long IFieldPos);
```

## 内部BASICシンタックス

```
Function AmGetFieldLongValue(hApiObject As Long, IFieldPos As Long) As Long
```

## 用途

バージョン：2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- `hApiObject`：クエリまたはレコードのハンドル
- `IFieldPos`：現在のオブジェクトに含まれているフィールドの番号

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

 注意:

この関数を使って日付、時刻、または日付+時刻のデータ型フィールドの値を取得する場合は、この関数が返す倍長整数は1970年1月1日午前0時0分0秒の時点からの秒数になります。

## AmGetFieldName()

現在のオブジェクトに含まれているフィールドの名前を返します。

## APIシンタックス

```
long AmGetFieldName(long hApiObject, long IFieldPos, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

```
Function AmGetFieldName(hApiObject As Long, IFieldPos As Long) As String
```

## 用途

バージョン : 2.52

使用

AssetCenter API



リンクまたはフィールドの設定スクリプト



	使用
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiObject : クエリ、レコード、またはテーブルのハンドル
- IFieldPos : 現在のオブジェクトに含まれているフィールドの番号。例えば、「0」という値は最初のフィールドを示します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldRights()

現在のオブジェクト内のフィールドに対するユーザ権限を返します。ユーザ権限は3つの文字のいずれかで返されます。3つの文字はそれぞれ、読取り、挿入、更新の権限を示します。

- 「r」はデータを読み取る権限を示します。
- 「i」はデータを挿入する権限を示します。
- 「u」はデータを更新する権限を示します。

例えば、フィールドのユーザ権限が読取り専用の場合は、「r」を返します。

## APIシンタックス

```
long AmGetFieldRights(long hApiObject, long IFieldPos, char *pstrBuffer, long IBuffer);
```

## 内部BASICシンタックス

Function AmGetFieldRights(hApiObject As Long, IFieldPos As Long) As String

### 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- hApiObject : クエリ、レコード、またはテーブルのハンドル
- IFieldPos : 現在のオブジェクトに含まれているフィールドの番号

### 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldSize()

フィールドのサイズを返します。

### APIシンタックス

```
long AmGetFieldSize(long hApiField);
```

## 内部BASICシンタックス

Function AmGetFieldSize(hApiField As Long) As Long

### 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiField : サイズを取得するフィールドのハンドル

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldSqlName()

ハンドルでフィールドを識別し、そのフィールドのSQL名を、文字列（String形式）で返します。

### APIシンタックス

long AmGetFieldSqlName(long hApiField, char \*pstrBuffer, long lBuffer);

## 内部BASICシンタックス

Function AmGetFieldSqlName(hApiField As Long) As String

### 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- hApiField : SQL名を取得するフィールドの有効なハンドル

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldStringValue()

現在のオブジェクトに含まれているフィールドの値を返します。この値は、文字列形式で返されます。

注意：この関数をAssetCenterのAPIを介して使用する場合は、さらにpszBufferとlBufferパラメータが必要となります。これらのパラメータはそれぞれ、取得した文字列を保管するバッファとして使用する文字列と、そのバッファのサイズを指定します。pszBuffer文字列は、書式化され（文字で埋められ）、lBuffer、

次に示すコードは不正確で、バッファとして使用する文字列のサイズの指定が正しくありません。

```
Dim strBuffer as String
Dim IRec as Long
Dim IBuffer as Long
IBuffer=20
IRec=AmGetFieldStrValue(1, 0, strBuffer, IBuffer)
```

次に正しいコードを示します。

```
Dim strBuffer as String
Dim IRec as Long
Dim IBuffer as Long
strBuffer=String(21, " ") ' バッファを21文字に設定 (" ")
IBuffer=20
IRec=AmGetFieldStrValue(1, 0, strBuffer, IBuffer)
```

String関数を使ってバッファ文字列を書式化する時に、パッド文字として「0」を使わないようにします。特に、この関数がループ内で同一の文字列をバッファとして使用する場合は、AmGetFieldStrValue関数を呼び出す前に、バッファのサイズを指定してください。

## APIシンタックス

```
long AmGetFieldStrValue(long hApiObject, long IFieldPos, char *pstrBuffer, long IBuffer);
```

## 内部BASICシンタックス

```
Function AmGetFieldStrValue(hApiObject As Long, IFieldPos As Long) As String
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiObject : クエリまたはレコードのハンドル
- IFieldPos : 現在のオブジェクトに含まれているフィールドの番号

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetFieldType()

フィールドのデータ型を返します。

## APIシンタックス

```
long AmGetFieldType(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetFieldType(hApiField As Long) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField : データ型を取得するフィールドのハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

 **注意:**

次の表に、AmGetFieldType関数がフィールドの各データ型について返す値を示します。

返す値	対応するフィールドのデータ型
0	Undefined
1	Byte
2	Short
3	Long
4	Float
5	Double
6	String
7	Time stamp
8	Bin
9	Blob
10	Date
11	Time
12	Memo

## AmGetFieldType()

この関数は、ハンドルで識別されるフィールドの "UserType" (参照: database.txt ファイル) を、倍長整数型で返します。フィールド用の有効な戻り値は次の通りです。

値	説明
0	デフォルト
1	数値
2	はい/いいえ
3	金額
4	日付
5	日付+時刻
7	システムリストデータ
8	リストデータ
10	パーセント

値	説明
11	経過時間
12	テーブルまたはフィールドのSQL名

リンク用の有効な戻り値は次の通りです。

値	説明
0	普通
1	コメント
2	画像
3	履歴
4	任意管理項目値

バージョン4.0.0までは、関数はリンク用に必ず0を返していました。AssetCenterバージョン4.1.0以降からは、関数は、リンク用に以下の値の1つを返します。

- 0: 普通
- 1: コメント
- 2: 画像
- 3: 履歴
- 5: スクリプト

## APIシンタックス

```
long AmGetFieldType(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetFieldType(hApiField As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	<input checked="" type="checkbox"/>
リンクまたはフィールドの設定スクリプト	<input checked="" type="checkbox"/>
「スクリプト」タイプのアクションの設定	<input checked="" type="checkbox"/>

## 使用

---

 ウィザードスクリプト



---

 ウィザードの「FINISH.DO」スクリプト
 

---



## パラメータ

- hApiField : 入力タイプを取得するフィールドの有効なハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetForeignKey()

ハンドルでリンクを識別し、そのリンクの外部キーのハンドルを取得します。

## APIシンタックス

```
long AmGetForeignKey(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetForeignKey(hApiField As Long) As Long
```

## 用途

バージョン : 3.00

## 使用

---

 AssetCenter API
 

---



	使用
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 処理するリンクのハンドル

## AmGetIndex()

クエリ、レコード、またはテーブルのハンドルからインデックスオブジェクトを作成し、そのオブジェクトのハンドルを返します。

## APIシンタックス

```
long AmGetIndex(long hApiTable, long IPos);
```

## 内部BASICシンタックス

```
Function AmGetIndex(hApiTable As Long, IPos As Long) As Long
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiTable : テーブルのハンドル
- IPos : テーブル内のインデックスの位置

## AmGetIndexCount()

hApiTableパラメータに指定したテーブルに含まれているインデックス数を返します。

## APIシンタックス

```
long AmGetIndexCount(long hApiTable);
```

## 内部BASICシンタックス

```
Function AmGetIndexCount(hApiTable As Long) As Long
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiTable : テーブルのハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetIndexField()

インデックス内のフィールドの位置（インデックスのIpos番目のフィールド）で識別されるフィールド上のハンドルを返します。

## APIシンタックス

```
long AmGetIndexField(long hApiIndex, long IPos);
```

## 内部BASICシンタックス

```
Function AmGetIndexField(hApiIndex As Long, IPos As Long) As Long
```

## 用途

バージョン：3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiIndex：操作に関連するインデックス上で有効なハンドル

- IPos : インデックス内でのフィールドの位置

## AmGetIndexFieldCount()

インデックスを構成するフィールドの数を返します。

### APIシンタックス

```
long AmGetIndexFieldCount(long hApiIndex);
```

### 内部BASICシンタックス

```
Function AmGetIndexFieldCount(hApiIndex As Long) As Long
```

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiIndex : 操作に関連するインデックス上で有効なハンドル

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetIndexFlags()

インデックスのパラメータを返します。

### APIシンタックス

```
long AmGetIndexFlags(long hApiIndex);
```

### 内部BASICシンタックス

```
Function AmGetIndexFlags(hApiIndex As Long) As Long
```

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiIndex : 操作に関連するインデックス上で有効なハンドル

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

関数に戻される値は、次の値の論理結合（OR）の結果です。

- 1: インデックスは重複する値を許可します。
- 2: インデックスはNull値を許可します。
- 4: インデックスは大文字小文字を区別しません。

関数が値「3」を戻す場合、インデックスは重複する値とNull値を許可すると推論できます（1 OR 2 = 3）。

## AmGetIndexName()

インデックスの名前を返します。

## APIシンタックス

```
long AmGetIndexName(long hApiIndex, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

```
Function AmGetIndexName(hApiIndex As Long) As String
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	

	使用
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiIndex : 名前を取得する必要があるインデックスの有効なハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetLink()

テーブルのハンドルからリンクオブジェクトを作成し、そのオブジェクトのハンドルを返します。

## APIシンタックス

```
long AmGetLink(long hApiTable, long IPos);
```

## 内部BASICシンタックス

```
Function AmGetLink(hApiTable As Long, IPos As Long) As Long
```

## 用途

バージョン : 3.02

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiTable : テーブルのハンドル
- IPos : オブジェクト内のリンクの位置 (インデックス)

## AmGetLinkCardinality()

1つのリンクで関連付け可能なリンク数を返します。

## APIシンタックス

long AmGetLinkCardinality(long hApiField);

## 内部BASICシンタックス

Function AmGetLinkCardinality(hApiField As Long) As Long

## 用途

バージョン : 3.5

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 関連付け可能なリンク数を取得するリンクのハンドル

## 戻りコード

- 1 : 1-1 (1対1) のリンクが可能です。
- 2 : 1-n (1対n) のリンクが可能です。

## AmGetLinkCount()

現在のテーブル内のリンク数を返します。

## APIシンタックス

```
long AmGetLinkCount(long hApiTable);
```

## 内部BASICシンタックス

```
Function AmGetLinkCount(hApiTable As Long) As Long
```

## 用途

バージョン : 3.02

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiTable : 有効なテーブルのハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetLinkDstField()

hApiFieldパラメータで定義したリンクのリンク先フィールド（外部キー）を返します。

## APIシンタックス

```
long AmGetLinkDstField(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetLinkDstField(hApiField As Long) As Long
```

## 用途

バージョン：3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField：処理するリンクのハンドル

## AmGetLinkFeatureValue()

「リンク」タイプの任意管理項目の値を返します。

### APIシンタックス

```
long AmGetLinkFeatureValue(long hApiObject, long IFieldPos, long IRecordId);
```

### 内部BASICシンタックス

```
Function AmGetLinkFeatureValue(hApiObject As Long, IFieldPos As Long, IRecordId As Long) As Long
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiObject : クエリまたはレコードのハンドル
- IFieldPos : 現在のオブジェクト内のフィールドの位置
- IRecordId : 値を取得する任意管理項目のレコードID

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim q as String
q = "Select fv_link, IEmplDeptId From amEmplDept Where IEmplDeptId = " & [IEmplDeptId]
Dim hq as Long
hq = amQueryCreate()
Dim IErr as Long
IErr = amQueryGet(hq, q)
Dim IId as Long
IId = amGetFieldLongValue(hq, 1)
amMsgBox("str: " & amGetFieldStrValue(hq, 0))
amMsgBox("int: " &
amGetFieldLongValue(hq,0))
amMsgBox("lnk: " & amGetLinkFeatureValue(hq,0,IId))
```

## AmGetLinkFromName()

名前からリンクオブジェクトを作成し、そのオブジェクトのハンドルを返します。

### APIシンタックス

```
long AmGetLinkFromName(long hApiTable, char *strName);
```

### 内部BASICシンタックス

```
Function AmGetLinkFromName(hApiTable As Long, strName As String) As Long
```

### 用途

バージョン : 3.02

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiTable : テーブルのハンドル
- strName : リンクのSQL名

## AmGetLinkType()

リンクのタイプを返します。

## APIシンタックス

```
long AmGetLinkType(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetLinkType(hApiField As Long) As Long
```

## 用途

バージョン : 3.02

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField : タイプを取得するリンクのハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetMainField()

指定したテーブルのメインフィールドに対応するフィールドオブジェクトを作成し、そのオブジェクトのハンドルを返します。

## APIシンタックス

```
long AmGetMainField(long hApiTable);
```

## 内部BASICシンタックス

```
Function AmGetMainField(hApiTable As Long) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiTable : 取得するメインフィールドを含むテーブルのハンドル

## AmGetMemoField()

この関数は、あるテーブルのメモタイプフィールドに対応するフィールドオブジェクトを作成します。関数はこの方法で作成されたフィールドにハンドルを返します。

## APIシンタックス

```
long AmGetMemoField(long hApiTable);
```

## 内部BASICシンタックス

```
Function AmGetMemoField(hApiTable As Long) As Long
```

## 用途

バージョン : 4.1.0

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiTable : テーブルのメモフィールドの検索する場合の、テーブル上のハンドル

## AmGetNextAssetPin()

AmGetNextAssetPin APIは、デバイス ( IAssetId ) で使用可能な次のピンを検索します。連続番号がピンを並べ替えます。ポートの方向 ( bPinPortDir ) に応じて、使用可能なピンは昇順 ( bPinPortDir = 0 ) または降順 ( bPinPortDir = 1 ) で並べ替えられます。

### APIシンタックス

```
long AmGetNextAssetPin(long hApiCnxBase, long IAssetId, long bPinPortDir, long iPrevPinSeq);
```

### 内部BASICシンタックス

```
Function AmGetNextAssetPin(IAssetId As Long, bPinPortDir As Long, iPrevPinSeq As Long) As Long
```

### 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IAssetId : デバイスのID
- bPinPortDir : 検索する方向
  - 0=昇順
  - 1=降順
- iPrevPinSeq

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetNextAssetPort()

AmGetNextAssetPort APIは、デバイス (IAssetId) で一定の用途 (IDutyId) を果たすまたは全く用途を果たさない、使用可能な次のポートを検索します。ポートのステータスは「使用可能」でなければなりません。プールのフラグは、ポートのユーザ側 (bCheckUser) またはホスト側 (bCheckHost) が検証されなければならないかどうかを指定します。プールのフラグが「真 (true)」の場合、API はユーザ値 (bUserAvail) またはホスト値 (bHostAvail) を比較します。ポートは連続番号で並べ替えられます。ポートの方向 (bPinPortDir) に応じて、使用可能なポートは昇順 (bPinPortDir = 0) または降順 (bPinPortDir = 1) で並べ替えられます。

## APIシンタックス

```
long AmGetNextAssetPort(long hApiCnxBase, long IAssetId, long ICabCnxTypeId, long IDutyId, long bCheckUser, long bCheckHost, long bUserAvail, long bHostAvail, long bPinPortDir, long iPrevPortSeq);
```

## 内部BASICシンタックス

```
Function AmGetNextAssetPort(IAssetId As Long, ICabCnxTypeId As Long, IDutyId As Long, bCheckUser As Long, bCheckHost As Long, bUserAvail As Long, bHostAvail As Long, bPinPortDir As Long, iPrevPortSeq As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IAssetId : 検索するデバイスのID
- ICabCnxTypeId : ポート用ケーブルの接続タイプのID
- IDutyId : ポートの用途のID
- bCheckUser : ユーザ側を検証するフラグ
- bCheckHost : ホスト側を検証するフラグ
- bUserAvail : 確認するユーザ側の使用可能ステータス
- bHostAvail : 確認するホスト側の使用可能ステータス
- bPinPortDir : 検索するピンの方向
  - 0=昇順
  - 1=降順
- iPrevPortSeq

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetNextCableBundle()

AmGetNextCableBundle APIは、ケーブル (ICableId) で一定の用途 (IDutyId) を果たすまたは全く用途を果たさない、使用可能な次のバンドルを検索します。バンドルのステータスは「使用可能」でなければなりません。ブールのフラグは、バンドルのユーザ側 (bCheckUser) またはホスト側 (bCheckHost) が検証され

なければならぬかどうかを指定します。ブールのフラグが「真 (true)」の場合、APIはユーザ値 (bUserAvail) またはホスト値 (bHostAvail) を比較します。

## APIシンタックス

```
long AmGetNextCableBundle(long hApiCnxBase, long ICableId, long IDutyId, long bCheckUser, long bCheckHost, long bUserAvail, long bHostAvail);
```

## 内部BASICシンタックス

```
Function AmGetNextCableBundle(ICableId As Long, IDutyId As Long, bCheckUser As Long, bCheckHost As Long, bUserAvail As Long, bHostAvail As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- ICableId : 検証するケーブルのID
- IDutyId : 検索する用途のID
- bCheckUser : このパラメータはユーザ側のバンドルの接続を確認するかどうかを指定します。
- bCheckHost : このパラメータはホスト側のバンドルの接続を確認するかどうかを指定します。
- bUserAvail : 検索するユーザ側の接続ステータス
- bHostAvail : 検索するホスト側の接続ステータス

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetNextCablePair()

AmGetNextCablePair APIは、ケーブル (ICableId) 内の一定のタイプ (IPairTypeId) の使用可能な次のケーブルペアを検索します。ペアはケーブルペアのIDで並べ替えられます。

## APIシンタックス

```
long AmGetNextCablePair(long hApiCnxBase, long ICableId, long IPairTypeId, long iStartPairSeq);
```

## 内部BASICシンタックス

```
Function AmGetNextCablePair(ICableId As Long, IPairTypeId As Long, iStartPairSeq As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- ICableId : 検索するケーブルのID
- IPairTypeId : 検索するケーブルペアのタイプ
- iStartPairSeq

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetNTDomains()

データベースに接続するユーザのドメインを取得します。

## APIシンタックス

```
long AmGetNTDomains(char *pstrDomains, long IDomains);
```

## 内部BASICシンタックス

```
Function AmGetNTDomains() As String
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	

## 使用

---

 ウィザードの「FINISH.DO」スクリプト
 

---



## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetNTMachinesInDomain()

あるドメインのコンピュータを1列に表示して（コンピュータ名はコンマで区切る）取得します。ドメインが空の場合、関数はERR\_CANCEL(2)を返しますが、関数の実行は中断されません。

## APIシンタックス

```
long AmGetNTMachinesInDomain(char *strDomain, char *pstrMachines, long lMachines, long bUseDC);
```

## 内部BASICシンタックス

```
Function AmGetNTMachinesInDomain(strDomain As String, bUseDC As Long) As String
```

## 用途

バージョン：4.00

## 使用

---

 AssetCenter API



---

 リンクまたはフィールドの設定スクリプト



---

 「スクリプト」タイプのアクションの設定
 

---



	使用
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strDomain : 検索するドメイン名

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetNTUsersInDomain()

ドメイン上のユーザのリストを取得します。リストは2つの列で返されます（login,fullname）。列は「|」で区切られ、行は「,」で区切られます。

## APIシンタックス

```
long AmGetNTUsersInDomain(char *strDomain, char *pstrUsers, long lUsers);
```

## 内部BASICシンタックス

```
Function AmGetNTUsersInDomain(strDomain As String) As String
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strDomain : 検索するドメイン名

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetPOLinePrice()

発注明細の価格を計算します。

## APIシンタックス

```
double AmGetPOLinePrice(long hApiCnxBase, long IPOrdLineId);
```

## 内部BASICシンタックス

```
Function AmGetPOLinePrice(IPOrdLineId As Long) As Double
```

## 用途

バージョン : 4.00

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IPOrdLineId : 発注明細のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetPOLinePriceCur()

発注明細に適用される通貨コードを見つけます。

## APIシンタックス

```
long AmGetPOLinePriceCur(long hApiCnxBase, long IPOrdLineId, char *pstrPrice, long IPrice);
```

## 内部BASICシンタックス

```
Function AmGetPOLinePriceCur(IPOrdLineId As Long) As String
```

## 用途

バージョン : 4.00

## 使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IPOrdLineId : 発注明細のID

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetPOLineReference()

発注明細に対応するカタログリファレンスの説明を取得します。

## APIシンタックス

```
long AmGetPOLineReference(long hApiCnxBase, long IPOrdLineId, char *pstrRef, long IRef);
```

## 内部BASICシンタックス

```
Function AmGetPOLineReference(IPOrdLineId As Long) As String
```

## 用途

バージョン : 4.00

## 使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IPOrdLineId : 発注明細のID

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetRecordFromMainId()

テーブルの主キーの値でレコードを識別し、そのレコードのID番号を返します。

## APIシンタックス

```
long AmGetRecordFromMainId(long hApiCnxBase, char *strTable, long IId);
```

## 内部BASICシンタックス

```
Function AmGetRecordFromMainId(strTable As String, IId As Long) As Long
```

## 用途

バージョン : 2.52

## 使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strTable : 処理するレコードを含んでいるテーブルのSQL名
- IId : レコードを含んでいるテーブルの主キーの値

## 注

この関数はテーブルが存在しない場合を除いて、レコードハンドルを返します。テーブル内のレコードが1つも指定されていないと、この関数に返されるハンドルを使用する関数を新規に実行するたびに、エラーが発生します。

## AmGetRecordHandle()

ハンドルでクエリを識別し、そのクエリで取得したレコードのハンドルを返します。このレコードは、データベースへの書込みに使用できません。この関数は、クエリにレコードの主キーが含まれていないと、機能しません。

## APIシンタックス

```
long AmGetRecordHandle(long hApiQuery);
```

## 内部BASICシンタックス

```
Function AmGetRecordHandle(hApiQuery As Long) As Long
```

## 用途

バージョン : 2.52

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiQuery : クエリオブジェクトの有効なハンドル

## AmGetRecordId()

ハンドルでレコードを識別し、そのレコードのID番号を返します。挿入中のレコードの場合、値は「0」です。

## APIシンタックス

```
long AmGetRecordId(long hApiRecord);
```

## 内部BASICシンタックス

```
Function AmGetRecordId(hApiRecord As Long) As Long
```

## 用途

バージョン : 2.52

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiRecord : ID番号を取得するレコードの有効なハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetRelDstField()

リンクのターゲットフィールドのハンドルを返します。

## APIシンタックス

```
long AmGetRelDstField(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetRelDstField(hApiField As Long) As Long
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField : 処理に関連するリンクで有効なハンドル

## AmGetRelSrcField()

リンクのソースフィールドのハンドルを返します。

## APIシンタックス

```
long AmGetRelSrcField(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetRelSrcField(hApiField As Long) As Long
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField : 処理に関連するリンクの有効なハンドル

## AmGetRelTable()

N対Nリンクの関係テーブルのハンドルを返します。

## APIシンタックス

```
long AmGetRelTable(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetRelTable(hApiField As Long) As Long
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 処理に関連するリンクの有効なハンドル

## 戻りコード

エラーが発生した場合に、無効なハンドル（ゼロ）を返します。

## AmGetReverseLink()

hApiFieldパラメータに含まれているハンドルで逆リンクを識別し、その逆リンクのハンドルを返します。

## APIシンタックス

```
long AmGetReverseLink(long hApiField);
```

## 内部BASICシンタックス

Function AmGetReverseLink(hApiField As Long) As Long

### 用途

バージョン : 3.02

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiField : 取得する逆リンクのリンクのハンドル

## AmGetSelfFromMainId()

指定したテーブル内のレコードの説明文字列を返します。

### APIシンタックス

```
long AmGetSelfFromMainId(long hApiCnxBase, char *strTableName, long lId, char *pstrRecordDesc, long lRecordDesc);
```

## 内部BASICシンタックス

Function AmGetSelfFromMainId(strTableName As String, lId As Long) As String

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strTableName : 処理するレコードを含んでいるテーブルのSQL名
- IId : 処理するID番号

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetSourceTable()

hApiFieldパラメータに指定されているリンクのソーステーブルのハンドルを返します。

## APIシンタックス

```
long AmGetSourceTable(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetSourceTable(hApiField As Long) As Long
```

## 用途

バージョン : 3.02

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField : 取得するソーステーブルのリンクの有効なハンドル

## 戻りコード

エラーが発生した場合に、無効なハンドル（ゼロ）を返します。

## AmGetTable()

現在接続しているテーブルを位置で識別し、そのテーブルのハンドルを返します。

## APIシンタックス

```
long AmGetTable(long hApiCnxBase, long IPos);
```

## 内部BASICシンタックス

```
Function AmGetTable(IPos As Long) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IPos : 現在接続しているテーブルの位置。値は「0」とAmGetTableCountの間になります。

## 戻りコード

エラーが発生した場合に、無効なハンドル（ゼロ）を返します。

## AmGetTableCount()

現在接続しているデータベースのテーブル数を返します。

## APIシンタックス

```
long AmGetTableCount(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmGetTableCount() As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	

	使用
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetTableDescription()

ハンドルでテーブルを識別し、そのテーブルの長い説明を、文字列（String形式）で返します。

## APIシンタックス

```
long AmGetTableDescription(long hApiTable, char *pstrDesc, long IDesc);
```

## 内部BASICシンタックス

```
Function AmGetTableDescription(hApiTable As Long) As String
```

## 用途

バージョン：3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓

	使用
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiTable : 取得する長い説明のテーブルの有効なハンドル

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetTableFromName()

現在接続しているテーブルをSQL名で識別し、そのテーブルのハンドルを返します。

## APIシンタックス

```
long AmGetTableFromName(long hApiCnxBase, char *strName);
```

## 内部BASICシンタックス

```
Function AmGetTableFromName(strName As String) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strName : ハンドルを取得するテーブルのSQL名

## 戻りコード

エラーが発生した場合に、無効なハンドル（ゼロ）を返します。

## AmGetTableLabel()

ハンドルでテーブルを識別し、そのテーブルのラベルを文字列（String形式）で返します。

## APIシンタックス

```
long AmGetTableLabel(long hApiTable, char *pstrLabel, long lLabel);
```

## 内部BASICシンタックス

```
Function AmGetTableLabel(hApiTable As Long) As String
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	

	使用
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiTable : 取得するラベルのテーブルの有効なハンドル

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetTableName()

テーブルのSQL名を文字列として返します。

## APIシンタックス

```
long AmGetTableName(long hApiTable, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

```
Function AmGetTableName(hApiTable As Long) As String
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiTable : 取得するSQL名のテーブルのハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetTableRights()

ハンドルでテーブルを識別し、そのテーブルのユーザ権限を文字列（String形式）で返します。返される文字列には、追加権限と削除権限を示す2つの文字のいずれか、または両方が含まれます。

- 「c」は、ユーザにそのテーブルでの追加権限があることを示します。
- 「d」は、ユーザにそのテーブルでの削除権限があることを示します。

つまり、

- 「c」は、ユーザがそのテーブルで追加権限しか使えないことを示します。
- 「cd」は、ユーザがそのテーブルで追加権限と削除権限の両方を使えることを示します。

## APIシンタックス

```
long AmGetTableRights(long hApiTable, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

Function AmGetTableRights(hApiTable As Long) As String

### 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiTable : ユーザ権限を取得するテーブルの有効なハンドル

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetTableSqlName()

ハンドルでテーブルを識別し、そのテーブルのSQL名を文字列（String形式）で返します。

### APIシンタックス

```
long AmGetTableSqlName(long hApiTable, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

Function AmGetTableSqlName(hApiTable As Long) As String

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiTable：取得するSQL名のテーブルの有効なハンドル

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetTargetTable()

リンク先テーブルのSQL名を返します。

### APIシンタックス

long AmGetTargetTable(long hApiField);

## 内部BASICシンタックス

Function AmGetTargetTable(hApiField As Long) As Long

### 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- hApiField : 処理するリンクのハンドル

### 戻りコード

エラーが発生した場合に、無効なハンドル（ゼロ）を返します。

## AmGetTrace()

AmGetTrace APIは、2ノード（IUserId、IHostId）間のトレースをケーブルリンクのテーブル内で検索します。トレースの方向（iTraceDir）は、トレースがユーザからホスト方向（iTraceDir=1）またはホストからユーザ方向（iTraceDir=0）であるかを識別します。トレースタイプ（iTraceType）は、トレースが接続（iTraceType=1）であるかまたは切断（iTraceType=2）であるかを指定します。完全トレースのインジケータ（bFullTrace）は、トレースが変更されたノードのみを含むか（bFullTrace=0）または完全トレースを含むか（bFullTrace=1）を指定します。

## APIシンタックス

```
long AmGetTrace(long hApiCnxBase, long IUserId, long IHostId, long iTraceDir, long iTraceType, long bFullTrace, char *pstrTrace, long lTrace);
```

## 内部BASICシンタックス

```
Function AmGetTrace(IUserId As Long, IHostId As Long, iTraceDir As Long, iTraceType As Long, bFullTrace As Long) As String
```

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IUserId：開始する接続リンクのID
- IHostId：終了する接続リンクのID
- iTraceDir：接続の方向
  - 0=ホストからユーザへの方向
  - 1=ユーザからホストへの方向
- iTraceType：接続のタイプ
  - 1=接続
  - 2=切断
- bFullTrace：このパラメータは不完全なトレースを無視し、トレース文字列全体を返すように指定します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetTraceFromHist()

AmGetTraceFromHist APIは、新規の接続と既存の接続を区別するために、トレースの処理を用いてトレース履歴から文字列を計算します。

## APIシンタックス

```
long AmGetTraceFromHist(long hApiCnxBase, long IProjTraceOutId, long iTraceDir,
char *strDelimiter, char *pstrTraceint, long ITraceint, long bUpdateFlag);
```

## 内部BASICシンタックス

```
Function AmGetTraceFromHist(IProjTraceOutId As Long, iTraceDir As Long, strDelimiter
As String, bUpdateFlag As Long) As String
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IProjTraceOutId : プロジェクトのトレースID
- iTraceDir : 接続の方向
  - 0=ホストからユーザへの方向
  - 1=ユーザからホストへの方向
- strDelimiter : 既存の接続と切断を示す文字列の区切り文字
- bUpdateFlag : このオプションパラメータはフィールド amCabTraceOut.TraceString を更新します。
  - 0=偽 ( false )
  - 1=真 ( true )

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmGetTypedLinkField()

hApiFieldパラメータに指定されているリンクのリンク先テーブルのSQL名を値として含んでいるフィールドのハンドルを返します。

## APIシンタックス

```
long AmGetTypedLinkField(long hApiField);
```

## 内部BASICシンタックス

```
Function AmGetTypedLinkField(hApiField As Long) As Long
```

## 用途

バージョン : 3.02

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 処理の最初でパラメータに入力されるリンクの有効なハンドル

## AmGetVersion()

AssetCenterのビルド番号を文字列で返します。

## APIシンタックス

```
long AmGetVersion(char *pstrBuf, long lBuf);
```

## 内部BASICシンタックス

```
Function AmGetVersion() As String
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓

	使用
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmHasAdminPrivilege()

接続しているユーザが管理者権限を持っている場合に、TRUE（0でない値）を返します。

## APIシンタックス

```
long AmHasAdminPrivilege(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmHasAdminPrivilege() As Long
```

## 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	

## 使用

---

 ウィザードの「FINISH.DO」スクリプト
 

---



## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmHasRelTable()

この関数では、リンクに関係テーブルがあるかないかをテストできます。

## APIシンタックス

```
long AmHasRelTable(long hApiField);
```

## 内部BASICシンタックス

```
Function AmHasRelTable(hApiField As Long) As Long
```

## 用途

バージョン : 3.5

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField : 処理に関連するリンクの有効なハンドル

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmImportDocument()

ファイルから書類を作成しインポートします。

## APIシンタックス

```
long AmImportDocument(long hApiCnxBase, long IDocObjId, char *strTableName, char *strFileName, char *strCategory, char *strDesignation);
```

## 内部BASICシンタックス

```
Function AmImportDocument(IDocObjId As Long, strTableName As String, strFileName As String, strCategory As String, strDesignation As String) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	

## 使用

## ウィザードの「FINISH.DO」スクリプト



## パラメータ

- IDocObjId : テーブル [ amDocument ] の [ IDocObjId ] フィールドに格納される値
- strTableName : [ amDocument ] テーブルの [ DocObjTable ] フィールドに格納される値。実際上、これは書類のリンク先レコードを含むテーブルのSQL名です。
- strFileName : インポートするファイルの名前
- strCategory : AssetCenterに表示される書類のカテゴリ
- strDesignation : AssetCenterに表示される書類の名前

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmIncrementLogLevel()

strMsgメッセージを履歴ウィンドウに表示し、ウィザードの最終ページでノードを作成します。

ノードには、次のメッセージがすべて表示されます。

## 内部BASICシンタックス

Function AmIncrementLogLevel(strMsg As String, iType As Long) As Long

## 用途

バージョン : 3.5

## 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- strMsg : 表示されるメッセージのテキスト
- iType : メッセージに関連付けるタイプを定義します。メッセージのタイプがエラーの場合は「1」、警告の場合は「2」、情報の場合は「4」の値になります。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmInsertRecord()

以前に作成したレコードをデータベースに挿入します。AmCreateRecord関数を使って作成したレコードのみを挿入できます。クエリを使ってアクセスしたレコードは挿入できません。

## APIシンタックス

```
long AmInsertRecord(long hApiRecord);
```

## 内部BASICシンタックス

```
Function AmInsertRecord(hApiRecord As Long) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiRecord : データベースに挿入するレコードのハンドル

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmInstantiateReqLine()

ある依頼明細のインスタンスを直接作成します。

## APIシンタックス

```
long AmInstantiateReqLine(long hApiCnxBase, long IReqLineId, long bFinal, long IOrderLineId, float fQty);
```

## 内部BASICシンタックス

```
Function AmInstantiateReqLine(IReqLineId As Long, bFinal As Long, IOrderLineId As Long, fQty As Single) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト 「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IRequestLineId：依頼明細のID
- bFinal：このパラメータでは割当を終了するかどうかを指定します。
- IPOrderLineId：発注明細のID
- fQty：インスタント化する数量

## 戻りコード

- 0：成功
- 0以外：エラーコード

## 注

依頼された要素を調達サイクルを通さずに作成します。bFinal=FALSEの場合、「納品待ち」ステータスの要素が作成されます。

## AmInstantiateRequest()

ある依頼の全内容を直接インスタント化します。

## APIシンタックス

```
long AmInstantiateRequest(long hApiCnxBase, long IRequestId, long IMulFactor);
```

## 内部BASICシンタックス

Function AmInstantiateRequest(IRequestId As Long, IMulFactor As Long) As Long

### 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IRequestId : 依頼のID
- IMulFactor : 実行するインスタント生成の数

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmlsConnected()

現在の接続が有効かどうかをテストします。

### APIシンタックス

long AmlsConnected(long hApiCnxBase);

## 用途

バージョン : 3.00

使用

---

AssetCenter API




---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト

---

ウィザードの「FINISH.DO」スクリプト

---

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmlsFieldForeignKey()

フィールドがデータベースの外部キーかどうかを識別します。

## APIシンタックス

```
long AmlsFieldForeignKey(long hApiField);
```

## 内部BASICシンタックス

```
Function AmlsFieldForeignKey(hApiField As Long) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiField : 識別するフィールドのハンドル

## 戻りコード

- 1 : フィールドは外部キーです。
- 0 : フィールドは外部キーではありません。

## AmlsFieldIndexed()

フィールドにインデックスが付いているかどうかを識別します。

## APIシンタックス

```
long AmlsFieldIndexed(long hApiField);
```

## 内部BASICシンタックス

```
Function AmlsFieldIndexed(hApiField As Long) As Long
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	

	使用
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 識別するフィールドのハンドル

## 戻りコード

- 1 : フィールドにインデックスが付いています。
- 0 : フィールドにインデックスは付いていません。

## AmlsFieldPrimaryKey()

フィールドがデータベースの主キーかどうかを識別します。

## APIシンタックス

```
long AmlsFieldPrimaryKey(long hApiField);
```

## 内部BASICシンタックス

```
Function AmlsFieldPrimaryKey(hApiField As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓

	使用
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 識別するフィールドのハンドル

## 戻りコード

- 1 : フィールドは主キーです。
- 0 : フィールドは主キーではありません。

## AmlsLink()

ハンドルでオブジェクトを識別し、そのオブジェクトがリンクかフィールドかを識別します。

## APIシンタックス

```
long AmlsLink(long hApiField);
```

## 内部BASICシンタックス

```
Function AmlsLink(hApiField As Long) As Long
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓

使用

ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiField : 処理するオブジェクトのハンドル

## 戻りコード

- 1 : オブジェクトはリンクです。
- 0 : オブジェクトはフィールドです。

## AmlsTypedLink()

ハンドルでオブジェクトを識別し、そのオブジェクトが入力されたリンクかどうかを識別します。

## APIシンタックス

```
long AmlsTypedLink(long hApiField);
```

## 内部BASICシンタックス

```
Function AmlsTypedLink(hApiField As Long) As Long
```

## 用途

バージョン : 3.02

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓

使用

ウィザードの「FINISH.DO」スクリプト



## パラメータ

- hApiField : 処理するオブジェクトのハンドル

## 戻りコード

- 1 : オブジェクトは入力されたリンクです。
- 0 : オブジェクトは入力されたリンクではありません。

## AmLastError()

該当する接続のコンテキスト内で最後に実行した関数によって作成された最後のエラーコードを返します。

## APIシンタックス

```
long AmLastError(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmLastError() As Long
```

## 用途

バージョン : 2.52

使用

AssetCenter API



リンクまたはフィールドの設定スクリプト



「スクリプト」タイプのアクションの設定



ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmLastErrorMsg()

現在の接続で発生した最後のエラーメッセージを返します。

## APIシンタックス

```
long AmLastErrorMsg(long hApiCnxBase, char *pstrBuffer, long lBuffer);
```

## 内部BASICシンタックス

```
Function AmLastErrorMsg() As String
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmListToString()

AmDbGetList関数で取得した文字列の結果を、AmDbGetString関数と同じ形式で表示される文字列に変換します。

### APIシンタックス

```
long AmListToString(char *return, long lreturn, char *strSource, char *strColSep, char *strLineSep, char *strIdSep);
```

### 内部BASICシンタックス

```
Function AmListToString(strSource As String, strColSep As String, strLineSep As String, strIdSep As String) As String
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strSource：変換する文字列
- strColSep：変換する文字列の列区切りに使用する文字

- strLineSep : 変換する文字列の行区切りに使用する文字
- strIdSep : 変換する文字列のID区切りに使用する文字

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmLog()

履歴ウィンドウにstrMessageメッセージを表示します。

## 内部BASICシンタックス

Function AmLog(strMessage As String, iLogType As Long) As Long

## 用途

バージョン : 3.00

### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト

---

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strMessage : 表示するメッセージのテキスト

- iLogType : メッセージに関連付けるタイプを定義します。メッセージのタイプがエラーの場合は「1」、警告の場合は「2」、情報の場合は「4」の値になります。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## 例

```
AmLog("メッセージの本文")
```

## AmLoginId()

接続しているユーザのIDを返します。

## APIシンタックス

```
long AmLoginId(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmLoginId() As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓



## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

接続しているユーザのIDをデータベース内のフィールドのデフォルト値として定義します。

```
RetVal=AmLoginId()
```

## AmLoginName()

接続しているユーザのログイン名を返します。

## APIシンタックス

```
long AmLoginName(long hApiCnxBase, char *return, long lreturn);
```

## 内部BASICシンタックス

```
Function AmLoginName() As String
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

接続しているユーザのログイン名をデータベース内のフィールドのデフォルト値として定義します。

```
RetVal=AmLoginName()
```

## AmMapSubReqLineAgent()

依頼明細と発注明細間のリンクを可能にします。

## APIシンタックス

```
long AmMapSubReqLineAgent(long hApiCnxBase, long IRequestLineId, long IPorderLineId);
```

## 内部BASICシンタックス

```
Function AmMapSubReqLineAgent(IRequestLineId As Long, IPorderLineId As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト 「スクリプト」タイプアクションの設定	
ウィザードスクリプト ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IRequestLineId：依頼明細のID
- IPorderLineId：発注明細のID

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmMoveCable()

AmMoveCable APIはケーブル (ICableId) を現在の場所からターゲット場所 (IToLocId) へ移動させます。プロジェクト (IProjectId) と作業指示 (IWorkOrderId) に値が入力されていると、ケーブルはコメント (strComment) と共にプロジェクトと作業指示へ追加されます。コメントはケーブルに実行されるアクション (例：AのケーブルをBへ移動) を説明します。

## APIシンタックス

```
long AmMoveCable(long hApiCnxBase, long ICableId, long IToLocId, long IProjectId, long IWorkOrderId, char *strComment);
```

## 内部BASICシンタックス

Function AmMoveCable(ICableId As Long, IToLocId As Long, IProjectId As Long, IWorkOrderId As Long, strComment As String) As Long

## 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- ICableId：移動させるケーブルのID
- IToLocId：ケーブルの新規の場所のID
- IProjectId：プロジェクトのID
- IWorkOrderId：作業指示のID
- strComment：作業指示に添付されるコメント

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmMoveDevice()

AmMoveDevice APIはデバイス (IDeviceId) を現在の場所からターゲット場所 (IToLocationId) へ移動させます。プロジェクト (IProjectId) と作業指示 (IWorkOrderId) に値が入力されていると、デバイスはコメント (strComment) と共にプロジェクトと作業指示へ追加されます。コメントはデバイスに実行されるアクション (例：AのデバイスをBへ移動) を説明します。

## APIシンタックス

```
long AmMoveDevice(long hApiCnxBase, long IDeviceId, long IToLocationId, long IProjectId, long IWorkOrderId, char *strComment);
```

## 内部BASICシンタックス

```
Function AmMoveDevice(IDeviceId As Long, IToLocationId As Long, IProjectId As Long, IWorkOrderId As Long, strComment As String) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IDeviceId：移動させられるデバイスのID
- IToLocationId：デバイスの新規場所のID
- IProjectId：プロジェクトのID
- IWorkOrderId：作業指示のID
- strComment：作業指示に添付されるコメント

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmMsgBox()

メッセージが入ったダイアログボックスを表示します。

### 内部BASICシンタックス

Function AmMsgBox(strMessage As String, IMode As Long) As Long

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- strMessage : ダイアログボックスに表示するメッセージ
- IMode : 表示されるダイアログボックスのタイプ (0 : OKボタン付きの単純ボックス、1 : OKとキャンセルボタン付きのボックス、2 : キャンセルボタン付きのボックス)

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

### 例

```
AmMsgBox("メッセージの本文")
```

## AmOpenConnection()

AssetCenterデータベース名からセッションを作成します。strDataSourceは有効なAssetCenterデータソース名でなければなりません（有効なAssetCenterデータベース接続は、AssetCenterのログインボックスに一覧表示されます）。

同じデータベースまたは異なるデータベースのどちらでも、複数の接続を開くことができます。

## APIシンタックス

```
long AmOpenConnection(char *strDataSource, char *strUser, char *strPwd);
```

## 用途

バージョン : 2.52

使用

---

AssetCenter API



---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト

---

ウィザードの「FINISH.DO」スクリプト

---

## パラメータ

- strDataSource : データソースの名前
- strUser : 接続するユーザ名
- strPwd : 指定したユーザのパスワード

## AmOpenScreen()

この関数はAssetCenterで画面を開きます。

## 内部BASICシンタックス

Function AmOpenScreen(strScreenId As String, strContext As String, strFilter As String, iMode As Long, strBindField As String) As Long

## 用途

バージョン : 4.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strScreenId : 開くシステム画面のビューやユーザ画面のビューのSQL名 (この優先順位で開きます)。
- strContext : このオプションパラメータには、画面を開く時にリスト内で選択されるレコードの識別子のリストを入力します。
- strFilter : 画面を開く時にリストに適用されるAQLフィルタ
- iMode : 画面を開く際のモード (参照、編集、など)。入力可能な値は : 0 (現在進行中のアクションなし)、1 (現在進行中のアクションなし)、2 (変更中)、3 (作成中)、4 (複製中)、5 (追加中)、6 (選択中)。
- strBindField : このパラメータでは、画面を開く際のフィルタやモード (リンクしている画面を開く、など) を選択できます。フィールドのSQL名、または現在のコンテキストを使用する場合はCurrentSrcChoice値を入力します。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmPagePath()

ウィザードの実行パス（参照したページのリスト）を含む文字列を返します。前に戻るジャンプは無視します。

### 内部BASICシンタックス

Function AmPagePath() As String

### 用途

バージョン：3.00

#### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmProgress()

パーセンテージ単位で進行状況を表わすインジケータをウィザードの最終ページに表示します。

## 内部BASICシンタックス

Function AmProgress(iProgress As Long) As Long

### 用途

バージョン : 3.00

#### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト 

### パラメータ

- iProgress : 進行状況インジケータのサイズを定義するのに使う、処理状況を示すパーセント (0 - 100)

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

### 例

```
AmProgress(85)
```

この例の場合は、85%の完了状況を示すインジケータが表示されます。

## AmQueryCreate()

現在の接続でクエリオブジェクトを作成します。作成したオブジェクトを使って、AQLステートメントをデータベースサーバに送信できます。

## APIシンタックス

```
long AmQueryCreate(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmQueryCreate() As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## AmQueryExec()

AQLクエリを実行します。クエリの最初の結果を返します。次の結果は AmQueryNext関数を使って取得できます。

送信したクエリがMemo型のフィールドを返す場合は、結果の文字数は255文字まで(半角の場合)に制限されます。

## APIシンタックス

```
long AmQueryExec(long hApiQuery, char *strQueryCommand);
```

## 内部BASICシンタックス

```
Function AmQueryExec(hApiQuery As Long, strQueryCommand As String) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiQuery : AQLステートメントの送信先となるオブジェクトの有効なハンドル
- strQueryCommand : AQLクエリの本文 (文字列)

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmQueryGet()

AQLクエリをカーソルなしで実行します (結果は1つのみ)。クエリの最初の結果のみを返します。

## APIシンタックス

```
long AmQueryGet(long hApiQuery, char *strQueryCommand);
```

## 内部BASICシンタックス

```
Function AmQueryGet(hApiQuery As Long, strQueryCommand As String) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiQuery : AQLステートメントの送信先となるオブジェクトの有効なハンドル
- strQueryCommand : AQLクエリの本文 (文字列)

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmQueryNext()

事前にAmQueryExec関数を使って実行したクエリの結果を返します。

## APIシンタックス

```
long AmQueryNext(long hApiQuery);
```

## 内部BASICシンタックス

```
Function AmQueryNext(hApiQuery As Long) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiQuery : AQLステートメントの送信先となるオブジェクトの有効なハンドル

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmQuerySetAddMainField()

クエリのモードを変更し、テーブルのメインフィールドが、返されるフィールドのリストに自動的に追加されるようにします。この種のクエリはNull識別子のレコードを返しません。

## APIシンタックス

```
long AmQuerySetAddMainField(long hApiQuery, long bAddMainField);
```

## 内部BASICシンタックス

```
Function AmQuerySetAddMainField(hApiQuery As Long, bAddMainField As Long) As Long
```

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiQuery : クエリオブジェクトで有効なハンドル
- bAddMainField : 以下の2つの値があります。
  - True : テーブルのメインフィールドが追加される。
  - False : テーブルのメインフィールドは追加されない。

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmQuerySetFullMemo()

デフォルトでは、関数AmQueryExecの実行時に、クエリはMemoタイプのフィールドを254文字で切り捨てます。この関数はクエリのモードを変更し、Memoフィールドを完全に回復します。

## APIシンタックス

```
long AmQuerySetFullMemo(long hApiQuery, long bFullMemo);
```

## 内部BASICシンタックス

Function AmQuerySetFullMemo(hApiQuery As Long, bFullMemo As Long) As Long

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiQuery : クエリオブジェクトで有効なハンドル
- bFullMemo : 以下の2つの値があります。
  - True : クエリはMemoフィールド全体を返す。
  - False : クエリはMemoフィールドを254文字で切り捨てる。

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmQueryStartTable()

ハンドルでクエリを識別し、そのクエリによって処理されるテーブルのハンドルを返します。

### APIシンタックス

long AmQueryStartTable(long hApiQuery);

## 内部BASICシンタックス

Function AmQueryStartTable(hApiQuery As Long) As Long

### 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- hApiQuery : クエリオブジェクトの有効なハンドル

### 戻りコード

エラーが発生した場合に、無効なハンドル（ゼロ）を返します。

## AmQueryStop()

ハンドルでクエリを識別し、そのクエリの実行を中断します。このクエリは、AmQueryExec関数を使って予め起動しておく必要があります。

## APIシンタックス

long AmQueryStop(long hApiQuery);

## 内部BASICシンタックス

Function AmQueryStop(hApiQuery As Long) As Long

## 用途

バージョン : 2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiQuery : クエリオブジェクトの有効なハンドル

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmReceiveAllPOLines()

1件の発注明細に含まれるすべての製品を受領します（一括受領されます）。

 注意:

受領明細は注文品を受領する手続きを行った時点で担当者によって作成されま  
す。従って、その前に受領明細のレコードにアクセスすることはできません。

## APIシンタックス

```
long AmReceiveAllPOLines(long hApiCnxBase, long IPOrdId, long IDelivId);
```

## 内部BASICシンタックス

```
Function AmReceiveAllPOLines(IPOrdId As Long, IDelivId As Long) As Long
```

## 用途

バージョン：3.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト 「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IPOrdId：受領する製品が含まれている発注明細のID
- IDelivId：発注明細中のすべての製品を受領するのに使用する受領伝票のID

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmReceivePOLine()

1件の発注明細の発注数のうち一部の数量だけを受領し（部分受領）、その受領明細のIDを返します。

### 注意:

受領明細は注文品を受領する手続きを行った時点で担当者によって作成されません。従って、その前に受領明細のレコードにアクセスすることはできません。

## APIシンタックス

```
long AmReceivePOLine(long hApiCnxBase, long IPOrdLineId, long IDelivId, float fQty);
```

## 内部BASICシンタックス

Function AmReceivePOLine(IPOrdLineId As Long, IDelivId As Long, fQty As Single)  
As Long

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IPOrdLineId : 受領する製品が含まれている発注明細のID
- IDelivId : 1件の発注明細の一部の数量の注文品を受領するのに使用する受領伝票のID
- fQty : 受領伝票に記入する受領数

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmRefreshAllCaches()

AssetCenterで使用しているキャッシュを更新します。

## APIシンタックス

```
long AmRefreshAllCaches(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmRefreshAllCaches() As Long
```

## 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmRefreshLabel()

AmRefreshLabel APIは、あるテーブル（strTableName）内のレコード（IMainId）のラベル文字列を更新します。

## APIシンタックス

```
long AmRefreshLabel(long hApiCnxBase, long IMainId, char *strTableName, char *pstrLabel, long ILabel);
```

## 内部BASICシンタックス

Function AmRefreshLabel(IMainId As Long, strTableName As String) As String

### 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IMainId：更新されるID
- strTableName：IMainIdに関連するテーブルの名前

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmRefreshProperty()

strVarNameプロパティをパラメータで識別し、そのプロパティの値を再評価します。プロパティがスクリプトを使用する場合は、もう一度そのスクリプトを実行します。

スクリプトを使わないプロパティの場合は、階層構造を更新します。

## 内部BASICシンタックス

Function AmRefreshProperty(strVarName As String) As Long

### 用途

バージョン：3.00

#### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



### パラメータ

- strVarName：再評価するウィザードのプロパティ名

### 戻りコード

- 0：成功
- 0以外：エラーコード

## AmRefreshTraceHist()

AmRefreshTraceHist APIはプロジェクトのトレースの完全な履歴を更新します。APIにはまた、「個々の」トレース履歴の項目を更新するオプションパラメータがあります。このパラメータがない場合は、トレースの全履歴が更新されます。

### APIシンタックス

long AmRefreshTraceHist(long hApiCnxBase, long lCabTraceOutId, long lTraceHistId);

## 内部BASICシンタックス

Function AmRefreshTraceHist(ICabTraceOutId As Long, ITraceHistId As Long) As Long

### 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- ICabTraceOutId : ケーブルトレースの説明のID
- ITraceHistId : このオプションパラメータはトレース履歴の「個々の」項目を更新します。

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmReleaseHandle()

オブジェクトのハンドルとサブハンドルを解放します。

### APIシンタックス

long AmReleaseHandle(long hApiObject);

## 内部BASICシンタックス

Function AmReleaseHandle(hApiObject As Long) As Long

### 用途

バージョン：2.52

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- hApiObject：処理するオブジェクトのハンドル

### 戻りコード

- 0：成功
- 0以外：エラーコード

## AmRemoveCable()

AmRemoveCable APIは、ケーブル (ICableId) を現在の場所から取り除きます。ケーブルのステータスは「使用不可能」になります。プロジェクト (IProjectId) と作業指示 (IWorkOrderId) に値が入力されていると、ケーブルがコメント (strComment) と共にプロジェクトと作業指示に追加されます。このコメントはケーブルに実行されるアクションを説明します (例：「ケーブルを現在の場所から取り除く」)。

## APIシンタックス

```
long AmRemoveCable(long hApiCnxBase, long ICableId, long IProjectId, long IWorkOrderId, char *strComment);
```

## 内部BASICシンタックス

```
Function AmRemoveCable(ICableId As Long, IProjectId As Long, IWorkOrderId As Long, strComment As String) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- ICableId：取り除くケーブルのID
- IProjectId：プロジェクトのID
- IWorkOrderId：作業指示のID
- strComment：作業指示に添付されるコメント

## 戻りコード

- 0：成功
- 0以外：エラーコード

## AmRemoveDevice()

AmRemoveDevice APIは、デバイス ( IDeviceId ) を現在の場所から取り除きます。デバイスのステータスは「使用不可能」になります。プロジェクト ( IProjectId ) と作業指示 ( IWorkOrderId ) に値が入力されていると、デバイスがコメント ( strComment ) と共にプロジェクトと作業指示に追加されます。このコメントはデバイスに実行されるアクションを説明します ( 例 : 「デバイスを現在の場所から取り除く」 )。

### APIシンタックス

```
long AmRemoveDevice(long hApiCnxBase, long IDeviceId, long IProjectId, long IWorkOrderId, char *strComment);
```

### 内部BASICシンタックス

```
Function AmRemoveDevice(IDeviceId As Long, IProjectId As Long, IWorkOrderId As Long, strComment As String) As Long
```

### 用途

バージョン : 4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- IDeviceId : 取り除くデバイスのID
- IProjectId : プロジェクトのID
- IWorkOrderId : 作業指示のID
- strComment : 作業指示に添付されるコメント

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmReturnAsset()

この関数では資産を返却できます。

## APIシンタックス

```
long AmReturnAsset(long hApiCnxBase, long IAstId, long IReturnId, long bCanMerge);
```

## 内部BASICシンタックス

```
Function AmReturnAsset(IAstId As Long, IReturnId As Long, bCanMerge As Long) As Long
```

## 用途

バージョン : 4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IAstId : 返却する資産のID
- IReturnId : 返却のID
- bCanMerge : このパラメータでは、返却を返却伝票に既存の明細と統合するかどうかを指定できます。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmReturnContract()

この関数では契約を返却できません。

## APIシンタックス

```
long AmReturnContract(long hApiCnxBase, long ICntrlId, long IReturnId, long bCanMerge);
```

## 内部BASICシンタックス

```
Function AmReturnContract(ICntrlId As Long, IReturnId As Long, bCanMerge As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- ICntrlId : 返却する契約のID
- IReturnId : 返却のID
- bCanMerge : このパラメータでは、返却を返却伝票に既存の明細と統合するかどうかを指定できます。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmReturnPortfolioItem()

この関数ではポートフォリオ品目を返却できます。

## APIシンタックス

```
long AmReturnPortfolioItem(long hApiCnxBase, long IPfId, float fQty, long IFromRecptLineId, long IReturnId, long bCanMerge);
```

## 内部BASICシンタックス

```
Function AmReturnPortfolioItem(IPfId As Long, fQty As Single, IFromRecptLineId As Long, IReturnId As Long, bCanMerge As Long) As Long
```

## 用途

バージョン : 4.00

使用

AssetCenter API



使用

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定 

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- IPfId : 返却するポートフォリオ品目のID
- fQty : 返却する数量（モデルの単位）
- IFromRecptLineId : 元の受領明細のID
- IReturnId : 返却明細のID
- bCanMerge : このパラメータでは、返却を返却伝票に既存の明細と統合するかどうかを指定できます。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmReturnTraining()

この関数では研修を返却できます。

## APIシンタックス

```
long AmReturnTraining(long hApiCnxBase, long ITrainingId, long IReturnId, long bCanMerge);
```

## 内部BASICシンタックス

Function AmReturnTraining(ITrainingId As Long, IReturnId As Long, bCanMerge As Long) As Long

### 用途

バージョン：4.00

	使用
AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- ITrainingId：返却する研修のID
- IReturnId：返却のID
- bCanMerge：このパラメータでは、返却を返却伝票に既存の明細と統合するかどうかを指定できます。

### 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmReturnWorkOrder()

この関数では作業指示を返却できます。

## APIシンタックス

```
long AmReturnWorkOrder(long hApiCnxBase, long IWOId, long IReturnId, long bCanMerge);
```

## 内部BASICシンタックス

```
Function AmReturnWorkOrder(IWOId As Long, IReturnId As Long, bCanMerge As Long) As Long
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- IWOId：返却する作業指示のID
- IReturnId：返却のID
- bCanMerge：このパラメータでは、返却を返却伝票に既存の明細と統合するかどうかを指定できます。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmRevCryptPassword()

暗号化されたパスワードを解読します。

### APIシンタックス

```
long AmRevCryptPassword(long hApiCnxBase, char *return, long lreturn, char
*strPassword);
```

### 内部BASICシンタックス

```
Function AmRevCryptPassword(strPassword As String) As String
```

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strPassword : 解読するパスワード

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmRgbColor()

strTextパラメータに対応する色のRGB値を返します。

### APIシンタックス

```
long AmRgbColor(char *strText);
```

### 内部BASICシンタックス

```
Function AmRgbColor(strText As String) As Long
```

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strText : 色の名前 :
  - White
  - LtGray
  - Gray
  - Dkgray
  - Black
  - Red
  - Green
  - Blue
  - Yellow

- Cyan
- Magenta
- Dkyellow
- Dkgreen
- Dkcyan
- Dkblue
- Dkmagenta
- Dkred

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmRollback()

処理開始（`AmStartTransaction`関数により実行します）を宣言する前に加えられた変更をすべてキャンセルします。

## APIシンタックス

```
long AmRollback(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmRollback() As Long
```

## 用途

バージョン : 2.52

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmSetFieldValue()

レコード内の1つのフィールドを変更します。データベースは更新しません。変更は、レコードの更新または挿入時、またはトランザクションのコミット時に実行されます。

## APIシンタックス

```
long AmSetFieldValue(long hApiRecord, char *strFieldName, long tmValue);
```

## 内部BASICシンタックス

```
Function AmSetFieldValue(hApiRecord As Long, strFieldName As String, tmValue As Date) As Long
```

## 用途

バージョン : 3.00

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	

使用

ウィザードの「FINISH.DO」スクリプト



## パラメータ

- hApiRecord : 変更するフィールドを含んでいるレコードのハンドル
- strFieldName : 変更するフィールドのSQL名
- tmValue : フィールドの新しい日付 (Date) 形式の値

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmSetFieldDoubleValue()

レコード内の1つのフィールドを変更します。データベースは更新しません。

## APIシンタックス

```
long AmSetFieldDoubleValue(long hApiRecord, char *strFieldName, double dValue);
```

## 内部BASICシンタックス

```
Function AmSetFieldDoubleValue(hApiRecord As Long, strFieldName As String, dValue As Double) As Long
```

## 用途

バージョン : 3.00

使用

AssetCenter API



リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定



使用

---

**ウィザードスクリプト**

---

**ウィザードの「FINISH.DO」スクリプト** 

---

## パラメータ

- hApiRecord : 変更するフィールドを含んでいるレコードのハンドル
- strFieldName : 変更するフィールドのSQL名
- dValue : フィールドの新しい倍精度 ( Double ) 形式の値

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmSetFieldLongValue()

レコード内の1つのフィールドを変更します。データベースは更新しません。日付、時刻、または日付+時刻の値を変更するには、1970年1月1日0時0分0秒から経過した秒数で新しい値を指定します。

## APIシンタックス

```
long AmSetFieldLongValue(long hApiRecord, char *strFieldName, long lValue);
```

## 内部BASICシンタックス

```
Function AmSetFieldLongValue(hApiRecord As Long, strFieldName As String, lValue As Long) As Long
```

## 用途

バージョン : 2.52

## 使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiRecord : 変更するフィールドのハンドル
- strFieldName : 変更するフィールドのSQL名
- lValue : フィールドの新しい値

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmSetFieldStrValue()

レコード内の1つのフィールドを変更します。データベースは更新しません。

## APIシンタックス

```
long AmSetFieldStrValue(long hApiRecord, char *strFieldName, char *strValue);
```

## 内部BASICシンタックス

```
Function AmSetFieldStrValue(hApiRecord As Long, strFieldName As String, strValue As String) As Long
```

## 用途

バージョン : 2.52

使用

AssetCenter API	✓
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- hApiRecord : 変更するフィールドを含んでいるレコードのハンドル
- strFieldName : 変更するフィールドのSQL名
- strValue : フィールドの新しい文字列 (String) 形式の値

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmSetLinkFeatureValue()

指定したレコードのリンクタイプの任意管理項目の値を設定します。

## APIシンタックス

```
long AmSetLinkFeatureValue(long hApiRecord, char *strFeatSqlName, char *strDstSelfValue, long IDstId);
```

## 内部BASICシンタックス

```
Function AmSetLinkFeatureValue(hApiRecord As Long, strFeatSqlName As String, strDstSelfValue As String, IDstId As Long) As Long
```

## 用途

バージョン : 3.00

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiRecord : リンクタイプの任意管理項目に関連付けられているレコードのID
- strFeatSqlName : 値を設定するリンクタイプの任意管理項目のSQL名。任意管理項目のSQL名には、必ず頭に「fv\_」が付きます。
- strDstSelfValue : レコードに表示される任意管理項目の値。IDstIdのIDを持つレコードの「Self」値です。無効または存在しない値を指定した場合は、データベースの整合性が壊れる可能性があります。
- IDstId : リンクタイプの任意管理項目のリンク先レコードのID

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## Am SetProperty()

名前でプロパティを識別し、そのプロパティの値を設定します。さらに、このプロパティの階層構造を更新します。

## 内部BASICシンタックス

Function Am SetProperty(strVarName As String, vValue As Variant) As Long

## 用途

バージョン : 3.00

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strVarName : 値を設定するプロパティの名前
- vValue : プロパティの新しい値

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmShowCableCrossConnect()

ケーブルのクロスコネクションの画面を表示します。

## 内部BASICシンタックス

Function AmShowCableCrossConnect(ICableId As Long) As Long

## 用途

バージョン : 4.00

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- ICableId : 演算に関連するケーブルのID

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmShowDeviceCrossConnect()

ケーブルデバイスのクロスコネクションの画面を表示します。

## 内部BASICシンタックス

Function AmShowDeviceCrossConnect(IDeviceId As Long) As Long

## 用途

バージョン : 4.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IDeviceId : 演算に関連するケーブルデバイスのID

## 戻りコード

- 0 : 成功

- 0以外：エラーコード

## AmSqlTextConst()

文字列を、クエリ内で使用可能になるように変換します。以下の演算た文字列上に実行されます。

- シングルクォーテーションマーク (') は二重 (ダブル) になります。
- シングルクォーテーションマークが文字列の最初と最後に追加されます。

## APIシンタックス

```
long AmSqlTextConst(char *return, long lreturn, char *str);
```

## 内部BASICシンタックス

```
Function AmSqlTextConst(str As String) As String
```

## 用途

バージョン：4.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- str：処理する文字列

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strReq as String
strReq="SELECT IEmplDeptId FROM amEmplDept WHERE Name=" & amSqlTextC
onst(strName)
```

変数strNameにはシングルクォーテーションマークが含まれていますが、このクエリは有効です。

## AmStartTransaction()

続行に関連付けられているデータベースで新しい処理を開始します。次のCommitまたはRollbackステートメントによって、データベースに加えられたすべての変更を有効にするかキャンセルするかを指定します。

## APIシンタックス

```
long AmStartTransaction(long hApiCnxBase);
```

## 内部BASICシンタックス

```
Function AmStartTransaction() As Long
```

## 用途

バージョン : 2.52

### 使用

---

AssetCenter API




---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定



使用

---

**ウィザードスクリプト**

---

**ウィザードの「FINISH.DO」スクリプト** 

---

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmStartup()

AssetCenterライブラリへの要求を初期化します。この関数は、他のどの関数よりも先に適用する必要があります。

## APIシンタックス

```
void AmStartup();
```

## 用途

バージョン : 2.52

使用

---

AssetCenter API

---

**リンクまたはフィールドの設定スクリプト**

---

**「スクリプト」タイプのアクションの設定**

---

**ウィザードスクリプト**

---

**ウィザードの「FINISH.DO」スクリプト**

---

## AmTableDesc()

テーブルのSQL名から、<テーブルの> (<テーブルのSQL名>) 形式の説明文字列を作成します。

## APIシンタックス

```
long AmTableDesc(long hApiCnxBase, char *return, long lreturn, char *strSqlName);
```

## 内部BASICシンタックス

```
Function AmTableDesc(strSqlName As String) As String
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strSqlName : 説明文字列を取得するテーブルのSQL名。このパラメータに無効なSQL名が含まれていると、疑問符( ? )を返します。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

資産テーブル (SQL名 : amAsset) の説明文字列を作成します。

```
AmTableDesc("amAsset")
```

次の説明文字列を返します。

```
Assets (amAsset)
```

## AmTaxRate()

税金のタイプ、税区分、および日付に応じて、税率を計算します。

### APIシンタックス

```
double AmTaxRate(char *strTaxRateName, long lTaxLocId, long tmDate, double dValue);
```

### 内部BASICシンタックス

```
Function AmTaxRate(strTaxRateName As String, lTaxLocId As Long, tmDate As Date,
dValue As Double) As Double
```

### 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strTaxRateName：税率の計算に使用する税金の種類 of SQL 名
- lTaxLocId：税金のタイプに対応する税区分のID番号
- tmDate：税率を計算する日付

- dValue : このパラメータは現在使用されていません。互換性の理由により保持されていますが、使用しないでください。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## AmUpdateDetail()

この関数はデータ入力ウィザードに使用します。ウィザードのコンテキスト（ウィザードを使って更新またはデータを入力するレコードのテーブル）をこの関数で明確に指定します。値に応じて、テーブルのフィールドやリンクを更新、またはデータを入力します。この関数は非モーダルのウィザードでは使用不可能です。

## 内部BASICシンタックス

Function AmUpdateDetail(strFieldName As String, varValue As Variant) As Long

## 用途

バージョン : 3.00

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- strFieldName : 更新する任意管理項目のSQL名
- varValue : フィールドに入力する新しい値

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmUpdateRecord()

レコードを更新します。

## APIシンタックス

```
long AmUpdateRecord(long hApiRecord);
```

## 内部BASICシンタックス

```
Function AmUpdateRecord(hApiRecord As Long) As Long
```

## 用途

バージョン : 2.52

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- hApiRecord : 更新するフィールドを含んでいるレコードのハンドル

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmValueOf()

この関数はウィザードで使います。strVarNameパラメータで特定したプロパティを返します。

## 内部BASICシンタックス

Function AmValueOf(strVarName As String) As Variant

## 用途

バージョン : 3.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strVarName : 値を取得するプロパティの名前

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

次の例では、「Page1.Label」プロパティの値が戻ります。

```
AmValueOf("Page1.Label")
```

この関数を使うと、処理したプロパティの従属文字列が壊れる可能性がありますので、注意してください。

## AmWizChain()

ウィザードAに含まれているウィザードBを実行します。ウィザードBが終了すると、ウィザードAに戻ります。

## 内部BASICシンタックス

Function AmWizChain(strWizSqlName As String) As Long

## 用途

バージョン：3.00

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strWizSqlName : 実行するウィザードのSQL名

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## AmWorkTimeSpanBetween()

2つの日付（開始日と終了日）で指定する作業期間の時間を秒単位で返します。  
計算には業務用カレンダーの情報が使われます。

## APIシンタックス

```
long AmWorkTimeSpanBetween(char *strCalendarSqlName, long tmEnd, long tmStart);
```

## 内部BASICシンタックス

```
Function AmWorkTimeSpanBetween(strCalendarSqlName As String, tmEnd As Date,
tmStart As Date) As Date
```

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strCalendarSqlName : 作業期間の時間の計算に使用する業務用カレンダーのSQL名。このパラメータを省略すると、計算には営業時間が反映されません。
- tmEnd : 計算する作業期間の終了日
- tmStart : 計算する作業期間の開始日

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

1998年9月1日午前8時から1998年9月24日午後7時までの作業時間を計算します。計算に使用する「Calendar\_Paris」というSQL名のカレンダーには次の営業時間が設定されています。

- 月曜日 - 木曜日 : 午前8時 - 正午、午後2時 - 午後6時
- 金曜日 : 午前8時 - 正午、午後2時 - 午後5時。

```
AmWorkTimeSpanBetween("Calendar_Paris", "1998/09/24 19:00:00", "1998/09/01 08:00:00")
```

この例では、507,600という値が戻ります。この値は2つの日付の間の作業時間を秒で表したものです。

## AppendOperand()

関数に渡されるパラメータに応じて文字列を連結します。文字列は次のように連結されます。

```
strExprstrOperatorstrOperand
```

## 内部BASICシンタックス

Function AppendOperand(strExpr As String, strOperator As String, strOperand As String)  
As String

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strExpr : 連結される文字列式
- strOperator : 連結する演算子
- strOperand : 連結するオペランド

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

### 注意:

strExprまたはstrOperandパラメータのいずれかを省略すると、strOperatorは使用されません。

## ApplyNewVals()

[ ListBox ] コントロール内の同じセルに同じ値を割り当てます。

## 内部BASICシンタックス

Function ApplyNewVals(strValues As String, strNewVals As String, strRows As String, strRowFormat As String) As String

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strValues : 処理する [ ListBox ] コントロールの値が含まれている文字列
- strNewVals : セルに割り当てる新しい値
- strRows : 処理する行のID。各IDをコンマ(,)で区切ります。
- strRowFormat : サプリストの書式化命令。各命令をパイプ文字(|)で区切ります。個々の命令には、strNewValsパラメータを含んでいる列の番号を指定します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Asc()

文字列の最初の文字のASCIIコード（数値）を返します。

## 内部BASICシンタックス

Function Asc(strAsc As String)

## 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strAsc：処理する文字列

## 例

```
Dim iCount as Integer
Dim strString as String
For iCount=Asc("A") To Asc("Z")
    strString = strString & Str(iCount)
Next iCount
RetVal=strString
```

## Atn()

数値のアークタンジェントを返します。単位はラジアンです。

## 内部BASICシンタックス

Function Atn(dValue As Double) As Double

## 用途

バージョン : 3.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- dValue : アークタンジェントを取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dPi as Double
Dim strString as String
dPi=4*Atn(1)
strString = Str(dPi)
RetVal=strString
```

## BasicToLocalDate()

BASIC形式の日付を文字列の日付（Windowsのコントロールパネルに表示される形式）に変換します。

## 内部BASICシンタックス

Function BasicToLocalDate(strDateBasic As String) As String

## 用途

バージョン : 3.5

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strDateBasic : 変換するBASIC形式の日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## BasicToLocalTime()

BASIC形式の時刻を文字列の時刻（Windowsのコントロールパネルに表示される形式）に変換します。

## 内部BASICシンタックス

Function BasicToLocalTime(strTimeBasic As String) As String

## 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strTimeBasic : 変換するBASIC形式の時刻

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## BasicToLocalTimeStamp()

BASIC形式の日付+時刻を文字列の日付+時刻（Windowsのコントロールパネルに表示される形式）に変換します。

## 内部BASICシンタックス

Function BasicToLocalTimeStamp(strTSBasic As String) As String

## 用途

バージョン : 3.5

### 使用

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strTSBasic : 変換するBASIC形式の日付+時刻

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Beep()

コンピュータで警告音を鳴らします。

## 内部BASICシンタックス

Function Beep()

## 用途

バージョン：3.00

### 使用

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## CDbl()

式を倍精度型 (Double型) に変換します。

### 内部BASICシンタックス

Function CDbl(dValue As Double) As Double

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- dValue : 変換する式

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数 (必要に応じてAmLastErrorMsg()関数も) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dNumber As Double
Dim iInteger as Integer
iInteger = 25
dNumber=Cdbl(iInteger)
RetVal=dNumber
```

## ChDir()

現在のディレクトリを変更します。

## 内部BASICシンタックス

Function ChDir(strDirectory As String)

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strDirectory : 新しいディレクトリ

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## ChDrive()

現在のドライブを変更します。

## 内部BASICシンタックス

Function ChDrive(strDrive As String)

## 用途

バージョン：3.00

### 使用

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strDrive：新しいドライブ名

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Chr()

iChrパラメータから渡されたASCIIコードに対応する文字列を返します。

### 内部BASICシンタックス

Function Chr(iChr As Long) As String

### 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- iChr : 文字のASCIIコード

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim iCount as Integer
Dim iIteration as Integer
Dim strMessage as String
Dim strLF as String
strLF=Chr(10)
For iIteration=1 To 2
  For iCount=Asc("A") To Asc("Z")
    strMessage=strMessage+Chr(iCount)
  Next iCount
  strMessage=strMessage+strLF
Next iIteration
RetVal=strMessage
```

## CInt()

指定した式を整数型に変換します。

## 内部BASICシンタックス

Function CInt(iValue As Long) As Long

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- iValue : 変換する式

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim iNumber As Integer
Dim dDouble as Double
dDouble = 25.24589
iNumber=CInt(dDouble)
RetVal=iNumber
```

## CLng()

指定した式を倍長整数型に変換します。

## 内部BASICシンタックス

Function CLng(IValue As Long) As Long

## 用途

バージョン : 3.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- IValue : 変換する式

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim INumber As Long
Dim iInteger as Integer
iInteger = 25
INumber=CLng(iInteger)
RetVal=INumber
```

## Cos()

数値のコサインを返します。単位はラジアンです。

## 内部BASICシンタックス

Function Cos(dValue As Double) As Double

## 用途

バージョン : 3.00

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト 

---

	使用
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- dValue : コサインを取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dCalc as Double
dCalc=Cos(150)
RetVal=dCalc
```

## CountOccurences()

1つの文字列内に特定の文字列が何個含まれているかをカウントします。

## 内部BASICシンタックス

```
Function CountOccurences(strSearched As String, strPattern As String, strEscChar As String) As Long
```

## 用途

バージョン : 3.5

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strSearched : 処理する文字列
- strPattern : strSearchedパラメータ内で検索する文字列
- strEscChar : エスケープ文字。関数がstrSearched文字列内でこの文字を検出すると、検索を中止します。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=CountOccurences("you|me|you,me|you", "you", ",") : "2"を返します。
MyStr=CountOccurences("you|me|you,me|you", "you", "|") : "1"を返します。
```

## CountValues()

1つの文字列に含まれる要素数をカウントします。区切り文字やエスケープ文字を区別しません。

### 内部BASICシンタックス

Function CountValues(strSearched As String, strSeparator As String, strEscChar As String) As Long

### 用途

バージョン : 3.5

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- strSearched : 処理する文字列
- strSeparator : 要素を区切る区切り文字
- strEscChar : エスケープ文字。この文字が区切り文字の前に付くと、その区切り文字は無視されます。

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=CountValues("you|me|you /me|you", "|", "/") : ' 4を返します。
MyStr=CountValues("you|me|you /me|you", "|", "") : ' 5を返します。
```

## CSng()

指定した式を浮動小数点数型（Float型）に変換します。

## 内部BASICシンタックス

Function CSng(fValue As Single) As Single

## 用途

バージョン : 3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- fValue : 変換する式

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dNumber As Double
Dim iInteger as Integer
iInteger = 25
dNumber=CSng(iInteger)
RetVal=dNumber
```

## CStr()

指定した式を文字列型に変換します。

## 内部BASICシンタックス

Function CStr(strValue As String) As String

## 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strValue：変換する式

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dNumber As Double
Dim strMessage as String
dNumber = 2,452873
strMessage=CStr(dNumber)
RetVal=strMessage
```

## CurDir()

現在のパスを返します。

## 内部BASICシンタックス

Function CurDir() As String

## 用途

バージョン：3.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## CVar()

指定した式を可変型（Variant型）に変換します。

## 内部BASICシンタックス

Function CVar(vValue As Variant) As Variant

## 用途

バージョン：3.00

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- vValue：変換する式

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaContext()

This function returns the value of a context identified by its name.

### 内部BASICシンタックス

Function DaContext(strField As String) As Variant

### 用途

バージョン : 1.0

#### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

### パラメータ

- strField: This parameter contains the name of the context for which you want to find the value.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

This example verifies that the environment variables can be recovered.

```
' Check that environment variables can be retrieved - this also retrieve all
' variables in one shot, improving performances
IRc = DaFirstEnv()

' Get useful environment variables and store them in context
if IRc <> 0 then
' Get environment variables
strProcArch   = DaGetEnv( "PROCESSOR_ARCHITECTURE" )
strProcId     = DaGetEnv( "PROCESSOR_IDENTIFIER" )
strProcLevel  = DaGetEnv( "PROCESSOR_LEVEL" )
strProcRevision = DaGetEnv( "PROCESSOR_REVISION" )
strProcCount  = DaGetEnv( "NUMBER_OF_PROCESSORS" )
strOs         = DaGetEnv( "OS" )
strComputer   = DaGetEnv( "COMPUTERNAME" )

' Check that computer name is the right one
if strComputer <> DaContext( "Computer.Name" ) then
  print "Warning: computer name does not look like the right one !"
end if

' Store them in context
DaSetContext "EnvInfo.ProcArch", strProcArch
DaSetContext "EnvInfo.ProcId", strProcId
DaSetContext "EnvInfo.ProcLevel", strProcLevel
DaSetContext "EnvInfo.ProcRevision", strProcRevision
DaSetContext "EnvInfo.ProcCount", strProcCount
DaSetContext "EnvInfo.Os", strOs
DaSetContext "EnvInfo.Computer", strComputer

' Dump
print DaDumpContext()
end if
```

## DaCopy()

This function copies a file, a set of files or a folder, locally. This function works in one of two ways:

- It copies a file locally to the deployment server: The source and the destination of the copy are both on the deployment server.
- It copies a file locally to the deployment target: The source and the destination of the copy are both on the deployment target.

The way in which it works is determined by the value of the file.on\_server option, set using the DaSetOption() function:

- If this option is set to 1, the copy concerns the deployment server.
- If this option is set to 0, the copy concerns the deployment target.

---

 **注意:**

For further information, refer to the description of the DaSetOption() function.

---



---

 **注意:**

This function carries out the same operations as the Copy files activity.

---

## 内部BASICシンタックス

Function DaCopy(strSource As String, strDest As String, strNameFilter As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- **strSource**: This parameter contains the destination path of the elements to copy. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local copy to the deployment server) or an absolute path (in the case of a local copy to the deployment target).
- **strDest**: Destination path of the elements to copy. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local copy to the deployment server) or an absolute path (in the case of a local copy to the deployment target).
- **strNameFilter**: This parameter contains wildcard characters, as used by DOS (\* and ?), used to filter the file names.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' copy user spreadsheet documents to a backup folder
dim l as long
l = DaSetOption( "file.recursive", 1 )
l = DaCopy( "c:\%USERPROFILE%", "c:\backup", "*.xls" )
```

## DaDbDeleteList()

This function deletes one or more records from the AssetCenter database.

## 内部BASICシンタックス

Function DaDbDeleteList(strTable As String, strList As String, strLineSep As String, strIdSep As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strTable: This parameter contains the SQL name of the table in which the records are inserted or updated.
- strList: This parameter contains the character string describing the record to be updated or inserted.

### 注意:

This parameter can be populated by the list obtained by the DbGetList function, or if the strIdStep parameter is empty, by a list of record identifiers.

- strLineSep: This parameter contains the character used as line separator in the character string contained in the strList parameter.
- strIdSep: This parameter contains the identifier of the record to be updated or inserted. If this parameter is null, the function considers that it is an insertion. If the value is not null, the function considers that it is an update of the record having this value as the identifier.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaDbGetList()

This function returns, as a list, the result of an AQL query.

 **注意:**

For further information on the AQL query language, refer to the chapter AQL in the manual entitled Advanced use.

## 内部BASICシンタックス

Function DaDbGetList(strQuery As String, strColSep As String, strLineSep As String, strIdSep As String) As String

## 用途

バージョン : 1.0

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- strQuery: This parameter contains the AQL query you want to execute.
- strColSep :This parameter contains the character use as column separator in the result given by the function.  
A column is equivalent to a field in a record returned by the query.
- strLineSep: This parameter contains the character used as line separator in the result returned by the function.  
A line is equivalent to a record returned by the query.
- strIdSep: This parameter contains the character used to separate record identifiers in the results given by the function.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strList As String
Dim IWorkgroup, IErr As Long

' Get list of computer ids in workgroup 'ToChange'
strList = DaDbGetList( "SELECT Workgroup FROM amComputer WHERE Workgroup='ToChange'", "|", "/", "@" )

' Replace all 'ToChange' workgroups by 'Changed'. InStr will return 0 if not found
IWorkgroup = InStr( 0, strList, "ToChange" )
Do While IWorkgroup > 0
  strList = Left$( strList, IWorkgroup - 1 ) + "Changed" + Mid$( strList, IWorkgroup + Len("ToChange"), Len(strList) )

  IWorkgroup = InStr( 0, strList, "ToChange" )
Loop

' Update computers
IErr = DaDbSetList( "amComputer", "Workgroup", strList, "|", "/", "@" )
```

## DaDbSetList()

This function updates or creates records in a table of the AssetCenter database.

 **注意:**

This function is the opposite of the DaDbGetList function.

## 内部BASICシンタックス

Function DaDbSetList(strTable As String, strFields As String, strList As String, strColSep As String, strLineSep As String, strIdSep As String) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- strTable: SQL name of the table concerned by the operation.
- strFields: This parameter contains the SQL name of the fields to be modified (in the case of the modification of an existing record) or populated (in the case of the creation of a new record). The character defined in the strColSep parameter is used as a separator for the SQL names of the fields.
- strList: list of the records concerned by the operation. Each record is separated by the strLineSep parameter. In the case of a modification (update of a record) the identifier of the record to modify is specified by appending the character contained in the strIdSep parameter, followed by the record identifier.
- strColSep: This parameter contains the character used as column separator (a column represents a field).
- strLineSep: This parameter contains the character used as a record separator (line separator).
- strIdSep: This parameter contains the character used to specify the identifier of the record to modify.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' Update name and workgroup of the computer '42' and insert a new computer named 'NewComputer' in computers table
' We could also use 'NewName|NewWorkgroup@0' to insert the new computer.

!Err = DaDbSetList( "amComputer", "Name|Workgroup", "UpdatedName|UpdatedWorkgroup@42/NewName|NewWorkgroup", "|", "/", "@" )
```

## DaDelete()

This function copies a file, a set of files or a folder, locally. This function works in one of two ways:

- Delete from the deployment server (from the file depot).
- Delete from the deployment target.

The way in which it works is determined by the value of the `file.on_server` option, set using the `DaSetOption()` function:

- If this option is set to 1, the deletion concerns the deployment server.
- If this option is set to 0, the deletion concerns the deployment target.

---

### 注意:

For further information, refer to the description of the `DaSetOption()` function.

---



---

### 注意:

This function carries out the same operations as the Delete files activity.

---

## 内部BASICシンタックス

Function `DaDelete(strToDelete As String, strNameFilter As String) As Long`

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strToDelete: This parameter contains the path of the files to delete. If you want to delete the elements from the deployment server, it is a relative path of the absolute path of the file depot. If you want to delete the elements from the deployment target, it is an absolute path.
- strNameFilter: This parameter contains wildcard characters, as used by DOS (\* and ?), used to filter the file names.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
'Remove temporary files from server depot
dim l as long
l = DaSetOption( "file.recursive", 1 )
l = DaSetOption( "file.on_server", 1 )
l = DaDelete( "", "*.tmp" )
```

## DaDownload()

This function copies a file, folder or set of files to the deployment server depot. The source files must be located on the deployment target.

You can also configure the behavior of this function by setting certain options using the DaSetOption() option. A full list of this option is included in the description of the DaSetOption() function.

### 注意:

This function carries out the same operations as the Download files activity.

## 内部BASICシンタックス

Function DaDownload(strSrcPath As String, strDstPath As String, strSrcNameFilter As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strSrcPath: This parameter contains the path of the files to copy to the depot of the deployment server. It is an absolute path on the deployment target. For example:  

c:\program files\antivirus\update\file.exe
- strDstPath: This parameter contains the path of the file copied on the deployment server. This path is relative to the absolute path of the deployment server. For example, if the absolute path of the file depot is:

```
c:\files\depot
```

and you want to copy a file to this location:

```
c:\files\depot\software\antivirus\update\file.exe
```

then this parameter will have the following value:

```
software\antivirus\update
```

- strSrcNameFilter: This parameter contains wildcard characters, as used by DOS (\* and ?), used to filter the file names.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
'get target specific path within file depot
dim path as string
path = "logs/" & DaContext( "Computer.Name" )

' retrieve a log file from target
dim l as long
l = DaDownload( "c:\program files\application\setup", path, "setup.log" )
```

## DaDumpContext()

This function returns a string containing all the defined contexts and is mainly used in debug mode.

## 内部BASICシンタックス

Function DaDumpContext() As String

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' Check that environment variables can be retrieved - this also retrieve all
' variables in one shot, improving performances
IRc = DaFirstEnv()

' Get usefull environment variables and store them in context
if IRc <> 0 then
' Get environment variables
strProcArch   = DaGetEnv( "PROCESSOR_ARCHITECTURE" )
strProcId    = DaGetEnv( "PROCESSOR_IDENTIFIER" )
strProcLevel  = DaGetEnv( "PROCESSOR_LEVEL" )
strProcRevision = DaGetEnv( "PROCESSOR_REVISION" )
strProcCount  = DaGetEnv( "NUMBER_OF_PROCESSORS" )
strOs        = DaGetEnv( "OS" )
strComputer  = DaGetEnv( "COMPUTERNAME" )

' Check that computer name is the right one
if strComputer <> DaContext( "Computer.Name" ) then
```

```

    print "Warning: computer name does not look like the right one !"
end if

' Store them in context
DaSetContext "EnvInfo.ProcArch", strProcArch
DaSetContext "EnvInfo.ProcId", strProcId
DaSetContext "EnvInfo.ProcLevel", strProcLevel
DaSetContext "EnvInfo.ProcRevision", strProcRevision
DaSetContext "EnvInfo.ProcCount", strProcCount
DaSetContext "EnvInfo.Os", strOs
DaSetContext "EnvInfo.Computer", strComputer

' Dump
print DaDumpContext()
end if

```

## DaExec()

This function execute a program stored on the deployment target.

You can also configure the behavior of this function by setting certain options using the DaSetOption() option. A full list of this option is included in the description of the DaSetOption() function.

### 注意:

This function performs the same operations as the Execute activity.

## 内部BASICシンタックス

Function DaExec(strCmd As String, strPath As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API

[リンクまたはフィールドの設定スクリプト](#)

## 使用

---

「スクリプト」タイプアクションの設定

---

ウィザードスクリプト

---


ウィザードの「FINISH.DO」スクリプト

---



## パラメータ

- strCmd: This parameter contains the name of the program to be executed. For example:

- strPath: This parameter contains the full path of the executable. In the case of the previous example:

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Log ipconfig information
!Err = DaSetOption( "exec.log_output", 1 )
!Err = DaExec( "ipconfig /all", "" )
```

## DaExecAction()

This function executes an action, identified by its name, on a table in the AssetCenter database.

**注意:**

This function supports a limited number of tables (workflows, agents, computers, etc.).

## 内部BASICシンタックス

Function DaExecAction(strAction As String, strTable As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- strAction: This parameter contains the SQL name of the action as defined in AssetCenter.
- strTable: This parameter contains the SQL name of the table on which the action is executed.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim IErr as Long
IErr = DaExecAction( "MyAction", "amComputer")
```

## DaExecuteActionByName()

This function executes an action, identified by its name, on a table and a record in the AssetCenter database.

## 内部BASICシンタックス

Function DaExecuteActionByName(strAction As String, strTable As String, IRecord As Long) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strAction: This parameter contains the SQL name of the action as defined in AssetCenter.
- strTable: This parameter contains the SQL name of the table on which the action is executed.
- IRecord: This parameter contains the identifier of the record in the table on which the action is executed.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

 注意:

If no identifier is specified, the record number linked to the table is used.

## 例

```
Dim lErr as Long
lErr = DaExecuteActionByName( "MyAction", "amComputer", 3)
```

## DaFileATime()

This function returns the date and time at which the file or folder were accessed.

## 内部BASICシンタックス

Function DaFileATime(strPath As String) As Date

## 用途

バージョン : 1.0

使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

## 使用

---

「スクリプト」タイプアクションの設定

---

ウィザードスクリプト



---

ウィザードの「FINISH.DO」スクリプト


## パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaFileCRC()

This function returns the CRC of a file (Cyclic Redundancy Check).

## 内部BASICシンタックス

Function DaFileCRC(strPath As String) As Long

## 用途

バージョン : 1.0

## 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

使用

---

**「スクリプト」タイプのアクションの設定**

---

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaFileCTime()

This function returns the date and time of creation of the file or folder.

## 内部BASICシンタックス

Function DaFileCTime(strPath As String) As Date

## 用途

バージョン : 1.0

使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

## 使用

---

「スクリプト」タイプアクションの設定

---

ウィザードスクリプト



---

ウィザードの「FINISH.DO」スクリプト


## パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaFileLanguage()

This function returns the language of the file (if available).

## 内部BASICシンタックス

Function DaFileLanguage(strPath As String) As String

## 用途

バージョン : 1.0

## 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

使用

---

**「スクリプト」タイプアクションの設定**

---

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaFileMTime()

This function returns the date and time at which the file or folder were modified.

## 内部BASICシンタックス

Function DaFileMTime(strPath As String) As Date

## 用途

バージョン : 1.0

使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

## 使用

「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaFileSize()

This function returns the size of a file in bits. If the size of the file is too large (i.e it exceeds a long), the returned value is -1.

## 内部BASICシンタックス

Function DaFileSize(strPath As String) As Long

## 用途

バージョン : 1.0

## 使用

---

 AssetCenter API
 

---

使用

リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaFileType()

This function returns the file type concerned by the operation.

## 内部BASICシンタックス

Function DaFileType(strPath As String) As String

## 用途

バージョン : 1.0

使用

AssetCenter API

## 使用

リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

The function returns one of the following values:

- "r" for a file,
- "d" for a directory,
- "D" for a physical drive.

## DaFileVersion()

This function returns the version of a file (if available).

## 内部BASICシンタックス

Function DaFileVersion(strPath As String) As String

## 用途

バージョン : 1.0

## 使用

AssetCenter API
リンクまたはフィールドの設定スクリプト
「スクリプト」タイプアクションの設定

使用

---

ウィザードスクリプト

---

ウィザードの「FINISH.DO」スクリプト

---



## パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaFind()

This function performs a recursive search of the files and folders and creates a list of the files it finds. The list is then scanned using the `DaFindNext()` function.

## 内部BASICシンタックス

Function `DaFind(strPath As String, strNameFilter As String, IDepth As Long) As Long`

## 用途

バージョン : 1.0

使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

## 使用

## 「スクリプト」タイプアクションの設定

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- strPath: This parameter contains the path of the search folder.
- strNameFilter: This parameter contains wildcard characters, as used by DOS (\* and ?), used to filter the search.
- IDepth: This parameter contains the depth of the recursive research, in number of sub-folders under the strPath folder.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' Builds a list of all bmp files
Dim IErr as Long
IErr = DaFind( "install", "*.bmp", 3)
```

## DaFindNext()

This function retrieves the name of the next file or folder in the list created by the DaFind() function. When the end of the list is reached, the function returns an empty character string.

## 内部BASICシンタックス

Function DaFindNext() As String

### 用途

バージョン：1.0

#### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
Dim strFileName as String
strFileName = DaFindNext()
If strFileName <> 0 Then
  Print strFileName
Else
  Print "Reached end of file list"
End If
```

## DaFirstEnv()

This function builds the list of environment variables available on the deployment target. This list can be browsed using the DaNextEnv() and DaGetEnv() functions.

### 内部BASICシンタックス

Function DaFirstEnv() As Long

### 用途

バージョン : 1.0

#### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
' Check that environment variables can be retrieved - this also retrieve all
' variables in one shot, improving performances
IRc = DaFirstEnv()

' Get usefull environment variables and store them in context
if IRc <> 0 then
```

```

' Get environment variables
strProcArch  = DaGetEnv( "PROCESSOR_ARCHITECTURE" )
strProclD    = DaGetEnv( "PROCESSOR_IDENTIFIER" )
strProcLevel = DaGetEnv( "PROCESSOR_LEVEL" )
strProcRevision = DaGetEnv( "PROCESSOR_REVISION" )
strProcCount = DaGetEnv( "NUMBER_OF_PROCESSORS" )
strOs        = DaGetEnv( "OS" )
strComputer  = DaGetEnv( "COMPUTERNAME" )

' Check that computer name is the right one
if strComputer <> DaContext( "Computer.Name" ) then
  print "Warning: computer name does not look like the right one !"
end if

' Store them in context
DaSetContext "EnvInfo.ProcArch", strProcArch
DaSetContext "EnvInfo.ProclD", strProclD
DaSetContext "EnvInfo.ProcLevel", strProcLevel
DaSetContext "EnvInfo.ProcRevision", strProcRevision
DaSetContext "EnvInfo.ProcCount", strProcCount
DaSetContext "EnvInfo.Os", strOs
DaSetContext "EnvInfo.Computer", strComputer

' Dump
print DaDumpContext()
end if

```

## DaGetEnv()

This function returns the value of an environment variable - contained in the list built using the DaFirstEnv() function - identified by its name.

### 内部BASICシンタックス

Function DaGetEnv(strEnv As String) As String

### 用途

バージョン : 1.0

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strEnv: This parameter contains the name of the environment variable as it is displayed by the operating system.

 ヒント:

To display the list of environment variables declared on your computer:

- 1 Right-click your computer icon,
- 2 Select the Advanced tab,
- 3 Click Environment variables

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' Check that environment variables can be retrieved - this also retrieve all
' variables in one shot, improving performances
IRc = DaFirstEnv()

' Get useful environment variables and store them in context
if IRc <> 0 then
' Get environment variables
strProcArch   = DaGetEnv( "PROCESSOR_ARCHITECTURE" )
strProcId     = DaGetEnv( "PROCESSOR_IDENTIFIER" )
```

```

strProcLevel = DaGetEnv( "PROCESSOR_LEVEL" )
strProcRevision = DaGetEnv( "PROCESSOR_REVISION" )
strProcCount = DaGetEnv( "NUMBER_OF_PROCESSORS" )
strOs = DaGetEnv( "OS" )
strComputer = DaGetEnv( "COMPUTERNAME" )

' Check that computer name is the right one
if strComputer <> DaContext( "Computer.Name" ) then
  print "Warning: computer name does not look like the right one !"
end if

' Store them in context
DaSetContext "EnvInfo.ProcArch", strProcArch
DaSetContext "EnvInfo.ProcId", strProcId
DaSetContext "EnvInfo.ProcLevel", strProcLevel
DaSetContext "EnvInfo.ProcRevision", strProcRevision
DaSetContext "EnvInfo.ProcCount", strProcCount
DaSetContext "EnvInfo.Os", strOs
DaSetContext "EnvInfo.Computer", strComputer

' Dump
print DaDumpContext()
end if

```

## DaGetFileInfo()

This function returns the properties of a file or folder. The properties in question can be recovered directly using the DaFile\* functions. It works in one of two ways:

- Operation on the deployment server (in the file depot)
- Operation on the deployment target

The way in which it works is determined by the value of the file.on\_server option, set using the DaSetOption() function:

- If this option is set to 1, the operation is on the deployment server
- If this option is set to 0, the operation is on the deployment target.

---

### 注意:

For further information, refer to the description of the DaSetOption() function.

---

## 内部BASICシンタックス

Function DaGetFileInfo(strPath As String) As Long

### 用途

バージョン : 1.0

#### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



### パラメータ

- strPath: This parameter contains the path of the file or folder concerned by the operation. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.

### 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Dalmpersonate()

This function enables you to impersonate a user on the deployment target by simulating that user. This function can be particularly useful to simulate an administrator connection in order to install certain software on the target, for example.

## 内部BASICシンタックス

Function Dalmpersonate(strUser As String, strPassword As String, strDomain As String)  
As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strUser: This parameter contains the name of the user for which you want to simulate a connection to the target.
- strPassword: This parameter contains the password of the user for which you want to simulate a connection to the target.
- strDomain: This parameter contains the domain of the user for which you want to simulate a connection to the target.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' Upload whole 'install' directory
IErr = DaUpload( "install", "c:/tmp/install", "" )

' Impersonate
if IErr = 0 then IErr = DaImpersonate( "user", "password", "domain " )

' Execute setup as user, synchronous, and log errors
if IErr = 0 then
  IDummyErr = DaSetOption( "exec.synchronous", 1 )
  IDummyErr = DaSetOption( "exec.log_output", 1 )
  IDummyErr = DaSetOption( "exec.log_error", 1 )
end if
if IErr = 0 then IErr = DaExec( "c:/tmp/install/setup -i", "c:/tmp/install" )

' On error, raise 'error' event
if IErr <> 0 then DaSetReturnValue "ErrorEvent"
```

## DaMkDir()

This function creates a folder. It works in one of two ways:

- Creation on the deployment server (in the file depot)
- creation on the deployment target

The way in which it works is determined by the value of the file.on\_server option, set using the DaSetOption() function:

- If this option is set to 1, the creation is on the deployment server
- If this option is set to 0, the creation is on the deployment target.

---

 注意:

For further information, refer to the description of the DaSetOption() function.

---

## 内部BASICシンタックス

Function DaMkDir(strDirectory As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strDirectory: This parameter contains the path of the folder to create. If the creation is carried out on the deployment server, it is the path relative to the absolute path of the file depot. If the creation is on the deployment target, it is an absolute path.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaMove()

This function moves a file, a set of files or a folder. This function works in one of two ways:

- It moves a file locally to the deployment server: The source and the destination of the move are both in the deployment server.
- It moves a file locally to the deployment target: The source and the destination of the move are both on the deployment target.

The way in which it works is determined by the value of the file.on\_server option, set using the DaSetOption() function:

- If this option is set to 1, the move is on the deployment server.

- If this option is set to 0, the move is on the deployment target.

 **注意:**

For further information, refer to the description of the DaSetOption() function.

 **注意:**

This function carries out the same operations as the Move files activity.

## 内部BASICシンタックス

Function DaMove(strSource As String, strDest As String, strNameFilter As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strSource: This parameter contains the source path of the elements to move. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local move to the deployment server) or an absolute path (in the case of a local move to the deployment target).
- strDest: This parameter contains the destination path of the elements to move. Depending on how the function operates, it is either a path relative to the file depot (in the case of a local move to the deployment server) or an absolute path (in the case of a local move to the deployment target).
- strNameFilter: This parameter contains wildcard characters, as used by DOS (\* and ?), used to filter the file names.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' move old report files before getting new ones
Dim l as Long
l = DaSetOption( "file.on_server", 1 );
l = DaMove( "reports/current", "reports/old", "" )
```

## DaNetIpFromName()

This function resolves the name of the target using the DNS method and returns the IP address of the target.

## 内部BASICシンタックス

Function DaNetIpFromName(strHost As String, pstrIP As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- strHost: This parameter contains the name of the host (deployment target).
- pstrIP: IP address of the target

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaNetNBTName()

This function queries the deployment target for related network information. This function is equivalent to the DOS command:

```
nbstat -A <IP address of the target>
```

## 内部BASICシンタックス

Function DaNetNBTName(strHost As String, ITimeout As Long, pstrComputerName As String, pstrGroup As String, pstrMAC As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strHost: This parameter contains the name of the host (deployment target).
- ITimeout: This parameter contains the duration after which the operation is considered to have failed, if the target has not responded. This duration is expressed in milliseconds.
- pstrComputerName: Name of the target
- pstrGroup: Network group of the target
- pstrMAC: MAC address of the target

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaNetPing()

This function pings the deployment target. This enables you to test whether a computer is responding on the network.

## 内部BASICシンタックス

Function DaNetPing(ITimeout As Long) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

---

使用

ウィザードの「FINISH.DO」スクリプト



## パラメータ

- ITimeout: This parameter contains the duration after which the operation is considered to have failed, if the target has not responded. This duration is expressed in milliseconds.

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim ILive As Long
' Ping with 1000ms timeout
ILive = DaNetPing( 1000 )
If ILive = 0 Then
    DaSetReturnValue "NOK"
End If
```

## DaNetWakeOnLan()

This function sends a wake on LAN signal to a deployment target.

 注意:

This function performs the same operations as the Wake on LAN activity.

## 内部BASICシンタックス

Function DaNetWakeOnLan(strMAC As String) As Long

### 用途

バージョン : 1.0

#### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

### パラメータ

- strMAC: This parameter contains the MAC address of the deployment target, in hexadecimal format. The MAC address is made up of six hexadecimal pairs, which may or may not be separated (with the "-" or ":" characters).

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNetWinAddressByName()

This function resolves the name of the target using the specified method. The function returns the IP address of the target.

## 内部BASICシンタックス

Function DaNetWinAddressByName(strHost As String, iNameSpace As Long, pstrIP As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strHost: This parameter contains the name of the host (deployment target).
- iNameSpace: This parameter specifies the resolution method to use. The possible values are the following:
  - 0: Uses the default resolution mode of the operating system
  - 1: Uses DNS resolution
  - 2: Uses WINS resolution
- pstrIP: This parameter contains the name of the host (deployment target).

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaNextEnv()

This function returns the name of the next environment variable in the list built using the DaFirstEnv() function. If the end of the list is reached, the function returns an empty character string.

## 内部BASICシンタックス

Function DaNextEnv() As String

### 用途

バージョン : 1.0

#### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
' retrieve and print all environment variables from target
'
dim IRc as long
dim VarName, VarValue as String

If DaFirstEnv() <> 0 Then
  Do
    ' get next variable
    VarName = DaNextEnv()
    If VarName <> "" Then
```

```

    VarValue = DaGetEnv( VarName )
    Print VarName & "=" & VarValue
Else
    Exit Do
End If
Loop
End If

```

## DaNTFileCopyTo()

This function copies a file from a source folder to a destination folder, both on the deployment target.

### 内部BASICシンタックス

Function DaNTFileCopyTo(strSrcPath As String, strSrc As String, strDst As String) As Long

### 用途

バージョン : 1.0

#### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

### パラメータ

- strSrcPath: This parameter contains the full path of the folder containing the file to copy.
- strSrc: This parameter contains the name of the file to copy.
- strDst: This parameter contains the full path of the destination folder of the file to copy.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTFileCreateDir()

This function recursively creates a folder on the deployment target.

## 内部BASICシンタックス

Function DaNTFileCreateDir(strPath As String) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strPath: This parameter contains the full path of the folder on the remote computer.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTFileDelete()

This function delete files on a deployment target.

### 内部BASICシンタックス

Function DaNTFileDelete(strFile As String) As Long

### 用途

バージョン : 1.0

#### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- strFile: Path of the file to be deleted.

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTFileDeleteDir()

This function deletes a folder and its contents on the deployment target.

### 内部BASICシンタックス

Function DaNTFileDeleteDir(strPathConst As String) As Long

## 用途

バージョン : 1.0

使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strPathConst: Path of the folder to be deleted.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTFileDirCopyTo()

This function copies a folder and its contents to a deployment target.

## 内部BASICシンタックス

Function DaNTFileDirCopyTo(strSrcPath As String, strDst As String) As Long

## 用途

バージョン : 1.0

使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	

## 使用

---

 ウィザードスクリプト
 

---




---

 ウィザードの「FINISH.DO」スクリプト
 

---



## パラメータ

- strSrcPath: This parameter contains the path of the folder on the server.
- strDst: This parameter contains the path of the destination folder on the deployment target.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTFileDirDownload()

This function copies a folder and its contents from a deployment target to the deployment server.

## 内部BASICシンタックス

Function DaNTFileDirDownload(strSrcPath As String, strDst As String) As Long

## 用途

バージョン : 1.0

## 使用

---

 AssetCenter API
 

---



---

 リンクまたはフィールドの設定スクリプト
 

---



---

 「スクリプト」タイプのアクションの設定
 

---



---

 ウィザードスクリプト
 

---




---

 ウィザードの「FINISH.DO」スクリプト
 

---



## パラメータ

- strSrcPath: This parameter contains the path of the folder on the deployment server.
- strDst: This parameter contains the path of the destination folder on the server. This path is relative to the file depot path.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTFileDirUpload()

This function copies a folder and its contents to a deployment target.

## 内部BASICシンタックス

Function DaNTFileDirUpload(strSrcPath As String, strDst As String) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- strSrcPath: This parameter contains the path of the folder on the deployment server. This path is relative to the path of the file depot.
- strDst : This parameter contains the path of the destination folder on the deployment target.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTFileDownload()

This function downloads a file from the deployment target.

## 内部BASICシンタックス

Function DaNTFileDownload(strSrcPath As String, strSrc As String, strDst As String)  
As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strSrcPath: This parameter contains the full path of the folder (on the target) containing the file to be downloaded.
- strSrc: This parameter contains the name of the file to download.
- strDst: This parameter contains the full path of the folder (on the server) to which the file is downloaded.

## 戻りコード

- 0 : 成功

- 0以外：エラーコード

## DaNTFileUpload()

This function uploads a file to the deployment target.

### 内部BASICシンタックス

Function DaNTFileUpload(strSrcPath As String, strSrc As String, strDst As String) As Long

### 用途

バージョン：1.0

#### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

### パラメータ

- strSrcPath: This parameter contains the full path of the folder (on the server) containing the file to be uploaded.
- strSrc: This parameter contains the name of the file to copy.
- strDst: This parameter contains the full path of the folder (on the target) to which the file is uploaded.

### 戻りコード

- 0：成功
- 0以外：エラーコード

## DaNTRegistryLMAddStringValue()

This function adds a REG\_SZ or REG\_EXPAND type entry for a key.

### 内部BASICシンタックス

Function DaNTRegistryLMAddStringValue(strKey As String, strString As String, strValue As String, bExpand As Long) As Long

### 用途

バージョン : 1.0

#### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

### パラメータ

- strKey: This parameter contains the full name of the key containing the entry to create.
- strString: This parameter contains the name of the entry to be created.
- strValue: This parameter contains the value of the entry to be created.
- bExpand: If this parameter has a value of 1, the function only processes REG\_EXPAND-type entries.

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTRegistryLMCreateKey()

This function creates recursively a key in the HKEY\_LOCAL\_MACHINE hive of the registry.

### 内部BASICシンタックス

Function DaNTRegistryLMCreateKey(strKey As String) As Long

### 用途

バージョン : 1.0

#### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

### パラメータ

- strKey: This parameter contains the full path of the key in the HKEY\_LOCAL\_MACHINE hive of the registry.

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTRegistryLMDeleteKey()

This function deletes a key fro the Registry of the deployment target. All the entries of the key are deleted.

## 内部BASICシンタックス

Function DaNTRegistryLMDeleteKey(strKey As String) As Long

### 用途

バージョン : 1.0

#### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

### パラメータ

- strKey: This parameter contains the full name of the key to be deleted.

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTRegistryLMDeleteValue()

This function deletes a key entry fro the Registry of the deployment target.

## 内部BASICシンタックス

Function DaNTRegistryLMDeleteValue(strKey As String, strString As String) As Long

### 用途

バージョン : 1.0

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strKey: This parameter contains the full name of the key containing the entry to delete.
- strString : ce parametre contient le nom de l'entree a effacer.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTRegistryLMGetLongValue()

This function returns the value of a REG\_DWORD type entry of a registry key on the deployment target.

## 内部BASICシンタックス

Function DaNTRegistryLMGetLongValue(strKey As String, strString As String) As Long

## 用途

バージョン : 1.0

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strKey: This parameter contains the full name of the key containing the entry for which you want to recover the value.
- strString: This parameter contains the name of the entry for which you want to recover the value.

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaNTRegistryLMGetStringValue()

This function returns the value of a REG\_DWORD or REG\_EXPAND type entry of a key from the Registry of the deployment target.

## 内部BASICシンタックス

Function DaNTRegistryLMGetStringValue(strKey As String, strString As String) As String

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strKey: This parameter contains the full name of the key containing the entry for which you want to recover the value.
- strString: This parameter contains the name of the entry for which you want to recover the value.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaNTRegistryLMSetLongValue()

This function sets the value of a REG\_DWORD type entry for a key. The entry must exist in the key.

## 内部BASICシンタックス

Function DaNTRegistryLMSetLongValue(strKey As String, strString As String, IValue As Long) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strKey: This parameter contains the full name of the key containing the entry for which you want to set the value.
- strString: This parameter contains the name of the entry for which you want to set the value.
- IValue: This parameter contains the value of the entry for which you want to set the value.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTRegistryLMSetStringValue()

This function sets the value of a REG\_SZ or REG\_EXPAND type entry for a key. The entry must exist in the key.

## 内部BASICシンタックス

Function DaNTRegistryLMSetStringValue(strKey As String, strString As String, strValue As String, bExpand As Long) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strKey: This parameter contains the full name of the key containing the entry for which you want to set the value.
- strString: This parameter contains the name of the entry for which you want to set the value.
- strValue: This parameter contains the value of the entry for which you want to set the value.
- bExpand: If this parameter has a value of 1, the function only processes REG\_EXPAND-type entries.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTServiceInstall()

This function installs an NT service on the deployment target.

## 内部BASICシンタックス

Function DaNTServiceInstall(strServiceName As String, strDisplayName As String, strBinaryFullName As String, bAutoStart As Long, bInteractWithDesktop As Long) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strServiceName: This parameter contains the internal name of the service, as it will be displayed in the properties of the service (accessible via the Properties shortcut menu).
- strDisplayName: This parameter contains the name of the service as it is displayed in the main list of services.
- strBinaryFullName: This parameter contains the full path of the service executable on the deployment target.
- bAutoStart: When this parameter has a value of 1, the service is started automatically.
- bInteractWithDesktop: When this parameter has a value of 1, the service can interact with the operating system's desktop (message sending and user dialog boxes).

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTServiceStart()

This function starts a named NT service on a deployment target.

## 内部BASICシンタックス

Function DaNTServiceStart(strServiceName As String) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strServiceName: This parameter contains the name of a service to start. The name used is the internal name of the service and not the name displayed in the operating system's services console. To locate the internal name:
  - 1 Right-click on the name of the service.
  - 2 Select the Properties menu.
  - 3 The Service name (internal name) is displayed in the General tab.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTServiceStatus()

This function returns the status of an NT service on the deployment target.

## 内部BASICシンタックス

Function DaNTServiceStatus(strServiceName As String) As String

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strServiceName: This parameter contains the internal name of the service, as it is displayed in the properties of the service (accessible via the Properties shortcut menu).

## 戻りコード

The function returns one of the following values, each one corresponding to a status for the service:

- NOT FOUND
- STOPPED
- START\_PENDING
- STOP\_PENDING
- RUNNING
- CONTINUE\_PENDING
- PAUSE\_PENDING
- PAUSED

## DaNTServiceStop()

This function stops the execution of an NT service named on the deployment target.

## 内部BASICシンタックス

Function DaNTServiceStop(strServiceName As String) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strServiceName: This parameter contains the name of the service to stop. The name used is the internal name of the service and not the name displayed in the operating system's services console. To locate the internal name:
  - 1 Right-click on the name of the service.
  - 2 Select the Properties menu.
  - 3 The Service name (internal name) is displayed in the General tab.

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTServiceUninstall()

This function uninstalls an NT service on the deployment target.

## 内部BASICシンタックス

Function DaNTServiceUninstall(strServiceName As String) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strServiceName: This parameter contains the internal name of the service, as it is displayed in the properties of the service (accessible via the Properties shortcut menu).

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTWMIExecMethod()

This function enables you to use a WMI method on an object or a class.

## 内部BASICシンタックス

Function DaNTWMIExecMethod(strObjectPath As String, strMethod As String, strInParams As String, pstrOutParam As String) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strObjectPath: Name of the object or the class.
- strMethodName of the method (CIMV2 namespace).
- strInParamsList of (input) parameters of the method
- pstrOutParam: Description of the parameters (output) of the method.

For example, the result of the Create method applied to the WIN32\_Process process will look like:

```
instance of __PARAMETERS
{
  ProcessId = 400;
  ReturnValue = 0;
};
```

 **注意:**

The character string that is returned contains line breaks.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
ICount = DaNTWMIGetInstanceCount( "Win32_Process" )
For I = ICount - 1 to 0 Step -1
  strName = DaNTWMIGetProperty( "Win32_Process.Name", I )
  If strName = "NOTEPAD.EXE" Then
    strPID = DaNTWMIGetProperty( "Win32_Process.Handle", I )
    IErr = DaNTWMIExecMethod( "Win32_Process.Handle=" & strPID & "", "Terminate", strDummy )
  End If
Next I
```

Ce script arrete l'ensemble des processus NOTEPAD.EXE grace a la methode Terminate.

## DaNTWMIExecQuery()

This function enables you to enumerate the WMI by using a WMI query in WQL language.

## 内部BASICシンタックス

Function DaNTWMIExecQuery(strQuery As String) As Long

### 用途

バージョン : 1.0

#### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

### パラメータ

- strQuery: WQL query

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

### 例

```
lErr = DaNTWMIExecQuery( "select * from Win32_NTLogEvent WHERE Logfile='Security' " )
While DaNTWMINextItem()
print DaNTWMIGetCurrentPropertyValue( "Message", i )
Wend
lErr = DaNTWMIResetEnumeration()
```

Ce script enumere l'ensemble des evenements du journal d'evenements de la securite NT et recherche l'evenement specifie.

## DaNTWMIGetCurrentArrayValue()

This function is used with the DANTWMIExecQuery function. It recovers the values of properties (table type) of the enumerated object.

The function returns the list of properties of the value, separated by a character defined in the strSeparator.

### 内部BASICシンタックス

Function DaNTWMIGetCurrentArrayValue(strProperty As String, strSeparator As String) As String

### 用途

バージョン : 1.0

#### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



### パラメータ

- strProperty: This parameter contains the name of property (of the WMI object) for which you want to recover the value.
- strSeparator: This parameter contains the character used as separator for the values of the property.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim IErr As Long IErr = DaNTWMIExecQuery( "SELECT * FROM
Win32_NetworkAdapterConfiguration WHERE IPEnabled=TRUE" ) While
DaNTWMINextItem() print DaNTWMIGetCurrentArrayValue( "IPAddress", ";" ) Wend
```

## DaNTWMIGetCurrentPropertyValue()

This function is used in relation to the DANTWMIExecQuery function. It recovers the property values of the enumerated object.

 **注意:**

Use the DaNTWMIGetCurrentArrayValue() function if you want to retrieve the values of an array property.

## 内部BASICシンタックス

```
Function DaNTWMIGetCurrentPropertyValue(strProperty As String) As Variant
```

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strProperty: This parameter contains the name of property (of the WMI object) for which you want to recover the value.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaNTWMIGetInstanceCount()

This function returns the number of objects of a given class (for example, the number of hard drives, CPUs, etc. of a computer).

## 内部BASICシンタックス

Function DaNTWMIGetInstanceCount(strShortName As String) As Long

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strShortName: This parameter contains the name of the object for which you want to count the number of instances. The available objects are listed in the table below:

### Property

---

Win32\_Processor

---

Win32\_Processor

---

Win32\_PhysicalMemory

---

Win32\_OperatingSystem

---

Win32\_DiskDrive

---

Win32\_LogicalDisk

---

Win32\_Operatingsystem

---

Win32\_Operatingsystem.BuildNumber

---

Win32\_NetworkAdapter

---

Win32\_NetworkAdapterConfiguration

---

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## 例

The following example returns the number of hard drives on the computer:

```
Dim iHDCount as Integer
iHDCount = DaNTWMIGetInstanceCount("Win32_LogicalDisk")
print "Number of hard drives = " & iHDCount
```

## DaNTWMIGetPropertyvalue()

This function recovers the value of a property from a WMI object.

## 内部BASICシンタックス

Function DaNTWMIGetPropertyvalue(strShortName As String, idx As Long) As Variant

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strShortName: This parameter contains the name of the property for which you want to recover the value. The syntax used is the following:

```
<object>.<property>
```

You can also use the aliases listed in the table below:

Alias	Referenced property
CPUType	Win32_Processor.Family
CPUInternal	Win32_Processor.Caption
CPU Speed	Win32_Processor.CurrentClockSpeed
MemorySize	Win32_PhysicalMemory.Capacity
OSMemorySize	Win32_OperatingSystem.TotalVisibleMemorySize
TotalDiskSize	Win32_DiskDrive.Size
DiskSize	Win32_LogicalDisk.Size
DiskFreeSpace	Win32_LogicalDisk.FreeSpace
OperatingSystem	Win32_Operatingsystem.Caption
OSServiceLevel	Win32_Operatingsystem.ServicePackMajorVersion
OSBuildNumber	Win32_Operatingsystem.BuildNumber
PhysAddress	Win32_NetworkAdapter.MACAddress
DNSHostName	Win32_NetworkAdapterConfiguration.DNSHostName
DNSDomain	Win32_NetworkAdapterConfiguration.DNSDomain

- idx: This parameter contains the index of the object concerned by the operation. If, for example, you have several hard drives, each of them will have an index number.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

The following example recovers the clock speed of the computer's CPU:

```
Dim vSpeed as Variant
vSpeed = DaNTWMIGetPropertyValue("CPUSpeed", 0)
print "Speed = " & vSpeed
```

## DaNTWMIGetTotalPropertiesValue()

This function returns the sum of the values of a property for all objects with this property. In this way you can recover the disk space available for all the logical drives of the computer.

## 内部BASICシンタックス

Function DaNTWMIGetTotalPropertiesValue(strShortName As String) As Variant

## 用途

バージョン : 1.0

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- strShortName: This parameter contains the name of the property for which you want to recover the value. The syntax used is the following:

```
<object>.<property>
```

You can also use the aliases listed in the table below:

Alias	Referenced property
CPUType	Win32_Processor.Family
CPUInternal	Win32_Processor.Caption
CPU Speed	Win32_Processor.CurrentClockSpeed
MemorySize	Win32_PhysicalMemory.Capacity
OSMemorySize	Win32_OperatingSystem.TotalVisibleMemorySize
TotalDiskSize	Win32_DiskDrive.Size
DiskSize	Win32_LogicalDisk.Size
DiskFreeSpace	Win32_LogicalDisk.FreeSpace
OperatingSystem	Win32_Operatingsystem.Caption
OSServiceLevel	Win32_Operatingsystem.ServicePackMajorVersion
OSBuildNumber	Win32_Operatingsystem.BuildNumber
PhysAddress	Win32_NetworkAdapter.MACAddress
DNSHostName	Win32_NetworkAdapterConfiguration.DNSHostName
DNSDomain	Win32_NetworkAdapterConfiguration.DNSDomain

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

The following example returns the total capacity of all the logical drives of the computer:

```
Dim vCapacity as Variant
vCapacity = DaNTWMIGetTotalPropertiesValue("WIN32_PhysicalMemory.Capacity")
```

```

}
print "Physical memory = " & vCapacity

```

## DaNTWMINextItem()

This function is used in relation to the DANTWMIExecQuery function. It enumerates the set of elements returned by a WQL query.

### 内部BASICシンタックス

Function DaNTWMINextItem() As Long

### 用途

バージョン : 1.0

#### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaNTWMIResetEnumeration()

This function is used in relation to the DANTWMIExecQuery function. It reinitiates the enumeration being opened and frees up the corresponding memory.

## 内部BASICシンタックス

Function DaNTWMIResetEnumeration() As Long

### 用途

バージョン : 1.0

使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプアクションの設定

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## DaRegCreateKey()

This function creates a key in the Registry of the deployment target.

## 内部BASICシンタックス

Function DaRegCreateKey(strKey As String) As Long

### 用途

バージョン : 1.0

使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプアクションの設定

---

## 使用

---

 ウィザードスクリプト



---

 ウィザードの「FINISH.DO」スクリプト


## パラメータ

- strKey: This parameter contains the full name of the key to be created.

Example:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Peregrine Systems\Automated Desktop
Administration 6.0.0
```

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaRegDeleteKey()

This function deletes a key from the Registry of the deployment target.

## 内部BASICシンタックス

Function DaRegDeleteKey(strKey As String) As Long

## 用途

バージョン : 1.0

## 使用

---

 AssetCenter API

---

 リンクまたはフィールドの設定スクリプト

使用

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト



---

ウィザードの「FINISH.DO」スクリプト


## パラメータ

- strKey: This parameter contains the full name of the key to be deleted.

HKEY\_LOCAL\_MACHINE\SOFTWARE\Peregrine Systems\Automated Desktop Administration\6.0.0

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaRegExec()

This function executes several operations (defined in one script) on the Registry of the target. The script uses the same syntax as the .reg files of the operating system.

 注意:

This function corresponds to the Registry type activity.

## 内部BASICシンタックス

Function DaRegExec(strScript As String) As Long

## 用途

バージョン : 1.0

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strScript: This parameters contains the script to execute on the Registry.

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaRegGetValue()

This function returns the value of a registry key from the deployment target.

## 内部BASICシンタックス

Function DaRegGetValue(strKey As String, strValue As String) As Variant

## 用途

バージョン : 1.0

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	

使用

ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## パラメータ

- strKey: This parameter contains the full name of the key concerned by the operation.
- strValue: This parameter contains the name of the entry of the key for which you want to recover the value.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaRegOutputValue()

This function recovers the value of the contents of DaRegOutput.

## 内部BASICシンタックス

Function DaRegOutputValue(strKey As String, strValue As String) As Variant

## 用途

バージョン：1.0

使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓

## 使用

---

 ウィザードの「FINISH.DO」スクリプト
 

---



## パラメータ

- strKey: This parameter contains the full name of the key concerned by the operation. If this parameter is empty, any key will be used.
- strValue: This parameter contains the string to convert. If the parameter is empty, the default value will be used.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaRegSetValue()

This function defines the value for a key entry in the Registry of the deployment target.

## 内部BASICシンタックス

Function DaRegSetValue(strKey As String, strValue As String, vValue As Variant) As Long

## 用途

バージョン : 1.0

## 使用

---

 AssetCenter API
 

---

 リンクまたはフィールドの設定スクリプト
 

---

 「スクリプト」タイプアクションの設定
 

---

使用

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- strKey: This parameter contains the full name of the key for which you want to set the value.
- strValue: This parameter contains the name of the entry to be defined.
- vValue: This parameter contains the value to assign to the entry. If you do not specify a value for this parameter, the current value for the entry is deleted.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaRegStrValue()

This function converts a value expressed as a variant to an expression compatible with the registry (as a string).

## 内部BASICシンタックス

Function DaRegStrValue(vValue As Variant) As String

## 用途

バージョン : 1.0

使用

AssetCenter API

## 使用

リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- vValue: This parameter contains the value to convert

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaRegVarValue()

This function converts a string type value to a variant.

## 内部BASICシンタックス

Function DaRegVarValue(strValue As String) As Variant

## 用途

バージョン : 1.0

## 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	

## パラメータ

- strValue: This parameter contains the string to convert.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaRename()

This function renames a file or a folder. It works in one of two ways:

- renaming on the deployment server (in the file depot)
- renaming of the deployment server

The way in which it works is determined by the value of the `file.on_serveroption`, set using the `DaSetOption()` function:

- If this option is set to 1, the renaming is on the deployment server.
- If this option is set to 0, the renaming is on the deployment target.

---

 **注意:**

For further information, refer to the description of the `DaSetOption()` function.

---

## 内部BASICシンタックス

Function `DaRename(strSource As String, strDest As String) As Long`

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strSource: This parameter contains the path of the source file or folder to rename. If the operation is carried out on the deployment server, it is a path relative to the absolute path of the file depot. If the operation is carried out on the deployment target, it is an absolute path.
- strDest: This parameter contains the new name of the file or folder if the form of a full path. If the operation is carried out on the deployment server, it is path relative to absolute path in the file depot. If the operation is carried out on the deployment target, it is an absolute path.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaReturnValue()

This function returns the name of the output event of the current activity of the deployment workflow.

## 内部BASICシンタックス

Function DaReturnValue() As String

### 用途

バージョン : 1.0

#### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
Set a return value if not already set
If DaReturnValue = "" Then
  SetDaReturnValue( "MyEvent" )
End If
```

## DaRmdir()

This function deletes a folder and its contents. It works in one of two ways:

- Deletion on the deployment server (in the file depot)
- deletion on the deployment target

This way in which it works is determined by the value of the file.on\_server option, set set using the DaSetOption() function:

- if this option is set to 1, the deletion is on the deployment server
- If this option is set to 0, the deletion is on the deployment target

 **注意:**

For further information, refer to the description of the DaSetOption() function.

## 内部BASICシンタックス

Function DaRmDir(strDirectory As String) As Long

### 用途

バージョン : 1.0

#### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strDirectory: This parameter contains the path of the folder to delete. If the deletion is carried out on the deployment server, it is the path relative to the absolute path of the file depot. If the deletion is on the deployment target, it is an absolute path.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaSendMail()

This function sends an e-mail. Since the operation is performed by the deployment server, all the configuration parameters associated with sending an e-mail must be configured on the server.

The following protocols are supported:

- SMTP: The addresses are as follows:

SMTP:[name@address.domain]

- MAPI: The addresses are as following:

MAPI:[name of the mail box]

- VIM: The addresses are as follows:

VIM:[name/domain]

- AM: internal messaging system (corresponding to the employees referenced in the database). The addresses are as follows:

AM:[login of the recipient]



**注意:**

In the case of multiple recipients, each address is separated by a comma (",").

## 内部BASICシンタックス

Function DaSendMail(strTo As String, strCc As String, strBcc As String, strSubject As String, strMessage As String) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- strTo: This parameter contains the list of addresses of recipients of a message.
- strCc: This parameter contains the list of addresses of recipients to receive copies of a message.
- strBcc: This parameter contains the list of addresses of recipients to receive blind carbon copies (they do not appear in the list of recipients) of a message.
- strSubject: This parameter contains the title of the message.
- strMessage: This parameter contains the body of the message.

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaSetContext()

This function defines a new context in the list of contexts available for a deployment workflow.

## 内部BASICシンタックス

Function DaSetContext(strField As String, vValue As Variant)

## 用途

バージョン : 1.0

### 使用

---

AssetCenter API

---

リンクまたはフィールドの設定スクリプト

---

「スクリプト」タイプのアクションの設定

---

ウィザードスクリプト 

---

---

## ウィザードの「FINISH.DO」スクリプト

---

### パラメータ

- strField: This parameter contains the name of the context you define.
- vValue: This parameter contains the value of the context that you define.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
' Check that environment variables can be retrieved - this also retrieve all
' variables in one shot, improving performances
IRc = DaFirstEnv()

' Get usefull environment variables and store them in context
if IRc <> 0 then
' Get environment variables
strProcArch   = DaGetEnv( "PROCESSOR_ARCHITECTURE" )
strProclD    = DaGetEnv( "PROCESSOR_IDENTIFIER" )
strProcLevel = DaGetEnv( "PROCESSOR_LEVEL" )
strProcRevision = DaGetEnv( "PROCESSOR_REVISION" )
strProcCount = DaGetEnv( "NUMBER_OF_PROCESSORS" )
strOs        = DaGetEnv( "OS" )
strComputer  = DaGetEnv( "COMPUTERNAME" )

' Check that computer name is the right one
if strComputer <> DaContext( "Computer.Name" ) then
    print "Warning: computer name does not look like the right one !"
end if
```

```
' Store them in context
DaSetContext "EnvInfo.ProcArch", strProcArch
DaSetContext "EnvInfo.ProcId", strProcId
DaSetContext "EnvInfo.ProcLevel", strProcLevel
DaSetContext "EnvInfo.ProcRevision", strProcRevision
DaSetContext "EnvInfo.ProcCount", strProcCount
DaSetContext "EnvInfo.Os", strOs
DaSetContext "EnvInfo.Computer", strComputer

' Dump
print DaDumpContext()
end if
```

## DaSetOption()

This function enables you to define options for certain other functions. The functions concerned are:

- All File management functions
- All Remote execution functions

## 内部BASICシンタックス

Function DaSetOption(strOption As String, IValue As Long) As Long

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strOption: This parameter contains the name of the option on which you want to intervene. The list of available options is as follows:
  - find.showdir: When you select this option, the DaFind() function includes the folders in the returned list.
  - file.resume: When you select this option, the functions that transfer files - DaUpload() and DaDownload() - attempt to continue the transfer from where it was when the transfer was interrupted.
  - file.mkdir: When you select this option, the File management functions authorize you to create folders.
  - file.override: When you select this option, the File management functions authorize you to overwrite files with the same name in the target.
  - file.recursive: When you select this option, the File management functions become recursive.
  - file.on\_server: When you select this option, the DaMove(), DaCopy() and DaDelete() functions operate on the file depot of the deployment server and not on the deployment target.
  - file.force: When this option is selected, the file is forced. For example, when associated to the DaDelete function, this option enables you to delete a file in read-only.
  - exec.synchronous: When you select this option, the execution of a program on the deployment target is synchronous. All later executions wait until the earlier executions have finished.
  - exec.log\_output: When you select this option, the messages sent by the program during its execution on the target are recorded in a log file.
  - exec.log\_error: When you select this option, the messages resulting from a program-execution error on the target are recorded in a log file.
  - exec.force\_visibility: When you execute a program with a graphical interface, the operating system decides whether or not to display this graphical interface. If this option is selected, the display is determined by the value of the exec.visibility option.
  - exec.visibility: When you select this option, the execution of a program on the target causes the program's graphical interface to be displayed on the target. This option is used in conjunction with the exec.force\_visibility option.
- IValue : This parameter can take one of two values:
  - 0: the option is not selected
  - 1: The option is selected

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' Further invocations of DaUpload will not be recursive
!Err = DaSetOption( 'upload.recursive', 0 )
```

## DaSetReturnValue()

This function enables you to change the event triggered at the end of the execution of a Basic Script type activity. The deployment workflow continues by using the transition associated with the event defined by the strValue parameter.

 注意:

The activity's output event is referred to by its name and not by its title.

## 内部BASICシンタックス

Function DaSetReturnValue(strValue As String)

## 用途

バージョン : 1.0

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト 

---

## ウィザードの「FINISH.DO」スクリプト

---



### パラメータ

- strValue: This parameter contains the name of the event triggered at the end of execution of the Basic script.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
dim errCode as long

' call a function that may return error code
errCode = DaFind( "c:/images", "*.gif" )

If errCode <> 0 Then
  DaSetReturnValue( "Error" )
End If
```

### Date()

現在のシステムの日付を返します。

### 内部BASICシンタックス

Function Date() As Date

## 用途

バージョン：3.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DateSerial()

`iYear`、`iMonth`、および`iDay`パラメータで指定した形式で日付型の値を返します。

## 内部BASICシンタックス

Function DateSerial(`iYear As Long`, `iMonth As Long`, `iDay As Long`) As Date

## 用途

バージョン：3.00

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプアクションの設定	✓

使用

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- iYear : 西暦。0 - 99の間の場合は、1900年 - 1999年までの年を表します。その他の年に関しては、4桁（1800など）で指定する必要があります。
- iMonth : 月
- iDay : 日

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

各パラメータに、それぞれ日、月、または年を表す数式を利用できます。

```
DateSerial(1999-10, 3-2, 15-8)
```

例えば、上記の例は次の値を返します。

```
1989/1/7
```

パラメータの値が予想される範囲（日付ならば1-31、月ならば1-12など）以外の値の場合、関数は空の日付を返します。

## DateValue()

「日付 + 時刻」の値の日付の部分を返します。

## 内部BASICシンタックス

Function DateValue(tmDate As Date) As Date

### 用途

バージョン : 3.00

#### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	✓
「スクリプト」タイプのアクションの設定	✓
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- tmDate : 「日付 + 時刻」形式の日付

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

次の例は、

```
DateValue ("1999/09/24 15:00:00")
```

次の値を返します。

```
1999/09/24
```

## DaTrackingDelete()

This function deletes a record in the tracking table.

### 内部BASICシンタックス

Function DaTrackingDelete(strDomain As String, strCategory As String, strSection As String, strName As String) As Long

### 用途

バージョン : 1.0

#### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- strDomain: This parameter contains the domain of the record in the tracking table.
- strCategory: This parameter contains the category of the record in the tracking table.
- strSection: This parameter contains the section of the record in the tracking table.
- strName: This parameter contains the name of the record in the tracking table.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaTrackingGet()

This function recovers the value of a field from the tracking table.

### 内部BASICシンタックス

Function DaTrackingGet(strDomain As String, strCategory As String, strSection As String, strName As String, strField As String) As String

### 用途

バージョン : 1.0

#### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- strDomain: This parameter contains the domain of the record in the tracking table.
- strCategory: This parameter contains the category of the record in the tracking table.
- strSection: This parameter contains the section of the record in the tracking table.
- strName: This parameter contains the name of the record in the tracking table.
- strField: This parameter contains the name of the field in the tracking table.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaTrackingSet()

This function populates a field in the tracking table. If the record does not exist, it is created.

### 内部BASICシンタックス

Function DaTrackingSet(strDomain As String, strCategory As String, strSection As String, strName As String, strField As String, strValue As String) As Long

### 用途

バージョン : 1.0

	使用
AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

### パラメータ

- strDomain: This parameter contains the domain of the record in the tracking table.
- strCategory: This parameter contains the category of the record in the tracking table.
- strSection: This parameter contains the section of the record in the tracking table.
- strName: This parameter contains the name of the record in the tracking table.
- strField: This parameter contains the name of the field in the tracking table.
- strValue: This parameter contains the value of the field in the tracking table.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaTrackingTest()

This function returns TRUE if there is a record with these properties in the tracking table (amDaTracking).

### 内部BASICシンタックス

Function DaTrackingTest(strDomain As String, strCategory As String, strSection As String, strName As String) As Long

### 用途

バージョン : 1.0

#### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	✓
ウィザードの「FINISH.DO」スクリプト	✓

### パラメータ

- strDomain: This parameter contains the domain of the record in the tracking table.
- strCategory: This parameter contains the category of the record in the tracking table.
- strSection: This parameter contains the section of the record in the tracking table.
- strName: This parameter contains the name of the record in the tracking table.

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## DaUpload()

This function copies a file, folder or set of files to the deployment server. The source files must be stored in the file depot of the deployment server.

You can also configure the behavior of this function by setting certain options using the `DaSetOption()` option. A full list of this option is included in the description of the `DaSetOption()` function.

### 注意:

This function carries out the same operations as the Upload files activity.

## 内部BASICシンタックス

Function `DaUpload(strSrcPath As String, strDstPath As String, strSrcNameFilter As String) As Long`

## 用途

バージョン : 1.0

### 使用

AssetCenter API	
リンクまたはフィールドの設定スクリプト	
「スクリプト」タイプのアクションの設定	
ウィザードスクリプト	
ウィザードの「FINISH.DO」スクリプト	

## パラメータ

- strSrcPath: This parameter contains the path of the files to copy on the deployment target. It is a relative path depending on the path of the file depot on the deployment server. For example, if the absolute path of the depot is:

```
c: /files /depot
```

and you want to copy a file with the following absolute path:

```
c: /files /depot /software /antivirus /update /file.exe
```

then this parameter will have the following value:

```
software /antivirus /update
```

- strDstPath: This parameter contains the absolute path of the files copied on the deployment target. In the previous example this is:
- ```
c: /program files /antivirus /update
```
- strSrcNameFilter: This parameter contains wildcard characters, as used by DOS (\* and ?), used to filter the file names.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' Upload whole 'install' directory
IErr = DaUpload( "install", "c:/tmp/install", "" )

' Impersonate
if IErr = 0 then IErr = DaImpersonate( "user", "password", "domain " )

' Execute setup as user, synchronous, and log errors
if IErr = 0 then
  IDummyErr = DaSetOption( "exec.synchronous", 1 )
  IDummyErr = DaSetOption( "exec.log_output", 1 )
```

```

IDummyErr = DaSetOption( "exec.log_error", 1 )
end if
if !Err = 0 then !Err = DaExec( "c:/tmp/install/setup -i", "c:/tmp/install" )

' On error, raise 'error' event
if !Err <> 0 then DaSetReturnValue "ErrorEvent"

```

## DaWait()

This function pause a deployment workflow for a period of time expressed in seconds.

## 内部BASICシンタックス

Function DaWait(f!SecDelay As Double)

## 用途

バージョン : 1.0

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト



## パラメータ

- f!SecDelay: This parameter defines the length of the pause in seconds.

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' Waits half a second, prints a message and waits 10 seconds
DaWait (0.5)
print "Waiting again..."
DaWait (10)
```

## Day()

tmDateパラメータの日付を返します。

## 内部BASICシンタックス

Function Day(tmDate As Date) As Long

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- tmDate：処理する「時刻+日付」形式のパラメータ

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strDay as String
strDay=Day(Date())
RetVal=strDay
```

## EscapeSeparators()

区切り文字の前にエスケープ文字を付けます。

## 内部BASICシンタックス

Function EscapeSeparators(strSource As String, strSeparators As String, strEscChar As String) As String

## 用途

バージョン : 3.5

### 使用

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strSource : 処理する文字列
- strSeparators : エスケープ文字を付ける区切り文字。複数の区切り文字を宣言する場合は、strEscCharパラメータで指定するエスケープ文字で区切る必要があります。
- strEscChar : エスケープ文字。strSeparatorsパラメータのすべての区切り文字の前にこのエスケープ文字が付けられます。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=EscapeSeparators("you|me|you,me|you", "|,", "/" ) : "you /me /you /me /you"を返します。
```

## ExeDir()

この関数は実行可能ファイルのフルパスを返します。

## 内部BASICシンタックス

Function ExeDir() As String

## 用途

バージョン : 3.60

## 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strPath as string
strPath=ExeDir()
```

## Exp()

数値のべき乗を返します。

## 内部BASICシンタックス

Function Exp(dValue As Double) As Double

## 用途

バージョン：3.00

## 使用

|                 |
|-----------------|
| AssetCenter API |
|-----------------|

|                        | 使用 |
|------------------------|----|
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dValue : べき乗を取得する数値。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim iSeed as Integer
iSeed = Int((10*Rnd)-5)
RetVal = Exp(iSeed)
```

## ExtractValue()

1つの文字列内で、区切り文字が前に付いていない値を取り出します。取り出した値は、その文字列から削除されます。この関数は、エスケープ文字も区別します。値を抽出する文字列に区切り文字が含まれていない場合は、その文字列全体が抽出され、元の文字列は完全に削除されます。

## 内部BASICシンタックス

```
Function ExtractValue(pstrData As String, strSeparator As String, strEscChar As String)
As String
```

## 用途

バージョン : 3.5

### 使用

| AssetCenter API        |   |
|------------------------|---|
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- pstrData : 処理する文字列
- strSeparator : 処理する文字列の区切り文字
- strEscChar : エスケープ文字。この文字が区切り文字の前に付くと、その区切り文字は無視されます。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=ExtractValue("you,me", ",", "/" ):"you"を返し、"me"を文字列に残します
◦
MyStr=ExtractValue(",you,me", ",", "/" ):""を返し、"you,me"を文字列に残します
◦
MyStr=ExtractValue("you", ",", "/" ):"you"を返し、""を文字列に残します。
MyStr=ExtractValue("you/,me", ",", "/" ):"you/,me"を返し、""を文字列に残します
◦
MyStr=ExtractValue("you/,me", ",", "" ):"you/"を返し、"me"を文字列に残します
```

```
RetVal=""
```

## FileCopy()

ファイルまたはフォルダをコピーします。

### 内部BASICシンタックス

Function FileCopy(strSource As String, strDest As String) As Long

### 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strSource：コピーするファイルまたはディレクトリのフルパス
- strDest：コピー先のファイルまたはディレクトリのフルパス

### 戻りコード

- 0：成功
- 0以外：エラーコード

## FileDateTime()

ファイルの時刻と日付を倍長整数型で返します。

### 内部BASICシンタックス

Function FileDateTime(strFileName As String) As Date

### 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strFileName : 処理するファイルのフルパス

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## FileExists()

ファイルが存在するかどうかテストします。

## 内部BASICシンタックス

Function FileExists(strFileName As String) As Long

### 用途

バージョン : 3.00

#### 使用

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strFileName : 存在するかテストするファイルの完全パス

### 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## FileLen()

ファイルのサイズを返します。

## 内部BASICシンタックス

Function FileLen(strFileName As String) As Long

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strFileName：処理するファイルのフルパス

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Fix()

数値の整数部分を返します（負数の場合は、その値の正の方向の一番近い整数を返します）。

## 内部BASICシンタックス

Function Fix(dValue As Double) As Long

## 用途

バージョン：3.00

## 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプアクションの設定     | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- dValue : 整数部分を取得する数値

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dSeed as Double
dSeed = (10*Rnd)-5
RetVal = Fix(dSeed)
```

## FormatDate()

strFormatパラメータの式に従って日付を書式化します。

## 内部BASICシンタックス

Function FormatDate(tmFormat As Date, strFormat As String) As String

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- tmFormat : 書式化する日付
- strFormat : 書式化命令が含まれている式

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

次の例は、日付を書式化するコードの例です。

```
Dim MyDate
MyDate="2000/03/14"
RetVal=FormatDate(MyDate, "dddd d mmmm yyyy") : "Tuesday 14 March 2000"を返します。
```

## FormatResString()

文字列内の変数\$1、\$2、\$3、\$4、および\$5を、strParamOne、strParamTwo、strParamThree、strParamFour、およびstrParamFiveパラメータに渡された文字列にそれぞれ置き換えます。

### 内部BASICシンタックス

Function FormatResString(strResString As String, strParamOne As String, strParamTwo As String, strParamThree As String, strParamFour As String, strParamFive As String) As String

### 用途

バージョン : 3.5

#### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

### パラメータ

- strResString : 処理する文字列
- strParamOne : 変数\$1を置き換える文字列
- strParamTwo : 変数\$2を置き換える文字列
- strParamThree : 変数\$3を置き換える文字列
- strParamFour : 変数\$4を置き換える文字列
- strParamFive : 変数\$5を置き換える文字列

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

次の例は、

```
FormatResString("I$1he$2you$3", "you", "we", "they")
```

"Iyouheweyouthey"を返します。

## FV()

定額の定期的な支払、および固定利率に基づいて計算した、実際の年間支払金額を返します。

## 内部BASICシンタックス

Function FV(dbIRate As Double, iNper As Long, dbIPmt As Double, dbIPV As Double, iType As Long) As Double

## 用途

バージョン : 3.00

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- `dblRate` : 支払日ごとの利率。例えば、年利が6%のローンの毎月の支払日の利率は、次のようになります。

```
0.06/12=0.005 または 0.5%
```

- `iNper` : 総支払回数
- `dblPmt` : 1回の支払金額。支払金額には一般的に元金と利子が含まれます。
- `dblPV` : 実際に支払わなければならない金額 (総額)
- `iType` : 支払期限を示します。次のいずれかの値になります。
  - 0 : 支払が後払い (期間内の最後) の場合
  - 1 : 支払が先払い (期間内の初め) の場合

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- `AssetCenter`内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数 (必要に応じて`AmLastErrorMsg()`関数も) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

### 注意:

- `Rate`と`Nper`パラメータの計算には同じ単位の支払額を使用する必要があります。
- 支払う金額 (`Pmt`パラメータの金額) はマイナスの数値、受け取る総額はプラスの数値で表わされます。

## GetListItem()

区切り文字で区切られている文字列のINb番目の文字列を返します。

## 内部BASICシンタックス

Function GetListItem(strFrom As String, strSep As String, INb As Long, strEscChar As String) As String

### 用途

バージョン : 3.5

#### 使用

| AssetCenter API        |   |
|------------------------|---|
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

### パラメータ

- strFrom : 処理する文字列
- strSep : 処理する文字列の区切り文字
- INb : 取得する文字列の位置
- strEscChar : エスケープ文字。この文字が区切り文字の前に付くと、その区切り文字は無視されます。

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

次に2つの例を示します。

```
GetListItem("this_is_a_test", "_", 2, "%")
```

"is"を返します。

```
GetListItem("this%_is_a_test", "_", 2, "%")
```

"a"を返します。

## Hex()

10進法のパラメータの16進法の値を返します。

## 内部BASICシンタックス

Function Hex(dValue As Double) As String

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dValue : 16進法の値を取得する10進法の数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Hour()

tmTimeパラメータの時間の部分を返します。

## 内部BASICシンタックス

Function Hour(tmTime As Date) As Long

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- tmTime：処理する「時刻+日付」形式のパラメータ

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strHour as String
strHour=Hour(Date())
RetVal=strHour
```

## InStr()

文字列内で、検索する文字列が最初に見つかった文字の位置を返します。

## 内部BASICシンタックス

Function InStr(iPosition As Long, strSource As String, strPattern As String) As Long

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- iPosition : 検索の開始点。このパラメータは必ず指定する必要があります。65,535よりも小さい有効な正の整数でなければなりません。
- strSource : 処理する文字列
- strPattern : 検索する文字列

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strSource as String
Dim strToSearch as String
Dim iPosition
strSource = "Good Bye"
strToSearch = "Bye"
iPosition = Instr(2, strSource, strToSearch)
RetVal=iPosition
```

## Int()

数値の整数部分を返します（負数の場合は、その値の負の方向の一番近い整数を返します）。

## 内部BASICシンタックス

Function Int(dValue As Double) As Long

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dValue : 整数部分を取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim iSeed as Integer
iSeed = Int((10*Rnd)-5)
RetVal = Abs(iSeed)
```

## IPMT()

年間支払金額のうち、指定した支払日の利子の金額を返します。

## 内部BASICシンタックス

Function IPMT(dbIRate As Double, iPer As Long, iNper As Long, dbIPV As Double, dbIFV As Double, iType As Long) As Double

## 用途

バージョン : 3.00

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト 

「スクリプト」タイプアクションの設定 

## 使用

ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- dbIRate：支払日ごとの利率。例えば、年利が6%のローンの毎月の支払日の利率は、次のようになります。

|                        |
|------------------------|
| 0.06/12=0.005 または 0.5% |
|------------------------|

- iPer：計算する期間。1からNperの数値の間で指定します。
- iNper：総支払回数
- dbIPV：実際に支払わなければならない金額（総額）
- dbIFV：支払が終了した時の差引残高または将来の支払用に確保する金額。一般に、特にローンを返済する場合は、このパラメータを「0」に設定します。実際、すべての支払を済ませるとローンの値はゼロになります。
- iType：支払期限を示します。次のいずれかの値になります。
- 0：支払が後払い（期間内の最後）の場合
- 1：支払が先払い（期間内の初め）の場合

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

### 注意:

- RateとNperパラメータの計算には同じ単位の支払額を使用する必要があります。
- 支払う金額（Pmtパラメータの金額）はマイナスの数値、受け取る総額はプラスの数値で表わされます。

## IsNumeric()

文字列に数値が含まれるかどうかを識別します。

### 内部BASICシンタックス

Function IsNumeric(strString As String) As Long

### 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strString : 解析する文字列

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Kill()

ファイルを削除します。

## 内部BASICシンタックス

Function Kill(strKilledFile As String) As Long

### 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strKilledFile : 処理するファイルのフルパス

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## LCCase()

文字列パラメータに含まれるすべての文字を小文字に変換して返します。

## 内部BASICシンタックス

Function LCCase(strString As String) As String

### 用途

バージョン : 3.00

## 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプアクションの設定     | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strString : 小文字に変換する文字列

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' This example uses the LTrim and RTrim functions to strip leading ' and trailing spaces
, respectively, from a string variable.
' It uses the Trim function alone to strip both types of spaces.
' LCase and UCase are also shown in this example as well as the use
' of nested function calls

Dim strString as String
Dim strTrimString as String
strString = " <-Trim-> " : ' Initialize string.
strTrimString = LTrim(strString) : ' strTrimString = "<-Trim-> ".
strTrimString = LCase(RTrim(strString)) : ' strTrimString = " <-trim->".
strTrimString = LTrim(RTrim(strString)) : ' strTrimString = "<-Trim->".
' Using the Trim function alone achieves the same result.
strTrimString = UCase(Trim(strString)) : ' strTrimString = "<-TRIM->".
RetVal= "|" & strTrimString & "|"
```

## Left()

文字列の左端から、iNumberで指定した数の文字を返します。

### 内部BASICシンタックス

Function Left(strString As String, iNumber As Long) As String

### 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strString：処理する文字列
- iNumber：返す文字数

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim lWord, strMsg, rWord, iPos : ' Declare variables.
strMsg = "Left() Test."
iPos = InStr(1, strMsg, " ") : ' スペースを探す
lWord = Left(strMsg, iPos - 1) : ' 左側の単語を取得
rWord = Right(strMsg, Len(strMsg) - iPos) : ' 右側の単語を取得
strMsg=rWord+lWord : ' 2つの単語を交換
RetVal=strMsg
```

## LeftPart()

strSepパラメータに指定されている区切り文字の左側の文字列を1つ取得します。  
 左から右に向かって区切り文字を探します。  
 bCaseSensitiveパラメータを使って、大文字と小文字を区別することもできます。

## 内部BASICシンタックス

Function LeftPart(strFrom As String, strSep As String, bCaseSensitive As Long) As String

## 用途

バージョン : 3.5

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strFrom : 処理する文字列
- strSep : 処理する文字列の区切り文字

- `bCaseSensitive` : このパラメータを使って、大文字と小文字を区別する (=1)、または区別しない (=0) かを指定します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

文字列"This\_is\_a\_test"に、`LeftPart`、`LeftPartFromRight`、`RightPart`、および`RightPartFromLeft`関数を使った例を示します。

```
LeftPart("This_is_a_test","_",0)
```

"This"を返します。

```
LeftPartFromRight("This_is_a_test","_",0)
```

"This\_is\_a"を返します。

```
RightPart("This_is_a_test","_",0)
```

"test"を返します。

```
RightPartFromLeft("This_is_a_test","_",0)
```

"is\_a\_test"を返します。

## LeftPartFromRight()

`strSep`パラメータに指定されている区切り文字の左側にある文字列を1つ取得します。

右から左に向かって区切り文字を探します。

`bCaseSensitive`パラメータを使って、大文字と小文字を区別することもできます。

## 内部BASICシンタックス

Function LeftPartFromRight(strFrom As String, strSep As String, bCaseSensitive As Long) As String

### 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strFrom : 処理する文字列
- strSep : 処理する文字列の区切り文字
- bCaseSensitive : このパラメータを使って、大文字と小文字を区別する (=1)、または区別しない (=0) かを指定します。

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

文字列"This\_is\_a\_test"に、LeftPart、LeftPartFromRight、RightPart、およびRightPartFromLeft関数を使った例を示します。

```
LeftPart("This_is_a_test","_",0)
```

"This"を返します。

```
LeftPartFromRight("This_is_a_test","_",0)
```

"This\_is\_a"を返します。

```
RightPart("This_is_a_test","_",0)
```

"test"を返します。

```
RightPartFromLeft("This_is_a_test","_",0)
```

"is\_a\_test"を返します。

## Len()

文字列または可変型データに含まれる文字数を返します。

## 内部BASICシンタックス

Function Len(vValue As Variant) As Long

## 用途

バージョン : 3.00

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト 

「スクリプト」タイプのアクションの設定 

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- vValue : 処理する可変型データ

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strTest as String
Dim iLength as Integer
strTest = "Peregrine Systems"
iLength = Len(strTest) : iLengthの値は17
RetVal=iLength
```

## LocalToBasicDate()

文字列形式の日付（Windowsのコントロールパネルに表示される形式）をBASIC形式の日付に変換します。

## 内部BASICシンタックス

Function LocalToBasicDate(strDateLocal As String) As String

## 用途

バージョン : 3.5

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strDateLocal : 変換する文字列形式の日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## LocalToBasicTime()

文字列形式の時刻（Windowsのコントロールパネルに表示される形式）をBASIC形式の時刻に変換します。

## 内部BASICシンタックス

Function LocalToBasicTime(strTimeLocal As String) As String

## 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strTimeLocal : 変換する文字列形式の時刻

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## LocalToBasicTimeStamp()

文字列形式の日付+時刻（Windowsのコントロールパネルに表示される形式）をBASIC形式の日付+時刻に変換します。

## 内部BASICシンタックス

Function LocalToBasicTimeStamp(strTSLocal As String) As String

## 用途

バージョン : 3.5

### 使用

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strTSLocal : 変換する文字列形式の日付+時刻

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## LocalToUTCDate()

「日付 + 時刻」フォーマットの日付をUTCフォーマット（タイムゾーンに関係しない）へ変換します。

## 内部BASICシンタックス

Function LocalToUTCDate(tmLocal As Date) As Date

## 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- tmLocal : 「日付 + 時刻」形式の日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Log()

数値の自然対数を返します。

## 内部BASICシンタックス

Function Log(dValue As Double) As Double

## 用途

バージョン：3.00

### 使用

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dValue：対数を取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dSeed as Double
dSeed = Int((10*Rnd)-5)
RetVal = Log(dSeed)
```

## LTrim()

文字列の先頭のスペースをすべて取り除きます。

## 内部BASICシンタックス

Function LTrim(strString As String) As String

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strString : 処理する文字列

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' This example uses the LTrim and RTrim functions to strip leading ' and trailing spaces
, respectively, from a string variable.
' It uses the Trim function alone to strip both types of spaces.
' LCase and UCase are also shown in this example as well as the use
' of nested function calls

Dim strString as String
Dim strTrimString as String
strString = " <-Trim-> " : ' Initialize string.
strTrimString = LTrim(strString) : ' strTrimString = "<-Trim-> ".
strTrimString = LCase(RTrim(strString)) : ' strTrimString = " <-trim-> ".
strTrimString = LTrim(RTrim(strString)) : ' strTrimString = "<-Trim-> ".
' Using the Trim function alone achieves the same result.
strTrimString = UCase(Trim(strString)) : ' strTrimString = "<-TRIM-> ".
RetVal= "|" & strTrimString & "|"
```

## MakeInvertBool()

ブール値の逆の値を返します（0は1、他の値はすべて0になります）。

## 内部BASICシンタックス

Function MakeInvertBool(IValue As Long) As Long

## 用途

バージョン : 3.5

### 使用

AssetCenter API

[リンクまたはフィールドの設定スクリプト](#) 

|                        | 使用 |
|------------------------|----|
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- IValue : 処理する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyValue
MyValue=MakeInvertBool(0) : ' 1を返します。
MyValue=MakeInvertBool(1) : ' 0を返します。
MyValue=MakeInvertBool(254) : ' 0を返します。
```

## Mid()

文字列内の一部分の文字列を返します。

## 内部BASICシンタックス

Function Mid(strString As String, iStart As Long, iLen As Long) As String

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strString : 処理する文字列
- iStart : strStringから取り出す文字列の開始位置
- iLen : 取り出す文字列の長さ

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strTest as String
strTest="One Two Three" : ' テスト文字列を定義
strTest=Mid(strTest,5,3) : ' strTest="Two"
RetVal=strTest
```

## Minute()

tmTimeパラメータに含まれている時刻の分の部分を返します。

## 内部BASICシンタックス

Function Minute(tmTime As Date) As Long

### 用途

バージョン：3.00

#### 使用

| AssetCenter API        |   |
|------------------------|---|
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

### パラメータ

- tmTime：処理する「時刻+日付」形式のパラメータ

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
Dim strMinute
strMinute=Minute(Date())
RetVal=strMinute : '現在の時刻の分の部分を返します。例えば、時刻が15:45:30の場合は、"45"を返します。
```

## MkDir()

新しいディレクトリを作成します。

### 内部BASICシンタックス

Function MkDir(strMkDirectory As String) As Long

### 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strMkDirectory : 作成するディレクトリのフルパス

### 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## Month()

tmDateパラメータに含まれている日付の月の部分を返します。

## 内部BASICシンタックス

Function Month(tmDate As Date) As Long

### 用途

バージョン：3.00

#### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

### パラメータ

- tmDate：処理する「時刻+日付」形式のパラメータ

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
Dim strMonth
strMonth=Month(Date())
RetVal=strMonth：'現在の月を返します。
```

## Name()

ファイルの名前を変更します。

### 内部BASICシンタックス

Function Name(strSource As String, strDest As String)

### 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strSource：名前を変更するファイルのフルパス
- strDest：新しいファイル名

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Now()

現在の日付と時刻を返します。

## 内部BASICシンタックス

Function Now() As Date

## 用途

バージョン : 3.00

### 使用

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## NPER()

定額の定期的な支払、および一定利率に基づく、年間の支払回数を返します。

## 内部BASICシンタックス

Function NPER(dblRate As Double, dblPmt As Double, dblPV As Double, dblFV As Double, iType As Long) As Double

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dblRate：支払日ごとの利率。例えば、年利が6%のローンの毎月の支払日の利率は、次のようになります。

|                          |
|--------------------------|
| $0.06/12=0.005$ または 0.5% |
|--------------------------|

- dblPmt：1回の支払金額。支払金額には一般的に元金と利子が含まれます。
- dblPV：実際に支払わなければならない金額（総額）
- dblFV：支払が終了したときの差引残高または将来の支払用に確保する金額。一般に、特にローンを返済する場合は、このパラメータを「0」に設定します。実際、すべての支払を済ませるとローンの値はゼロになります。
- iType：支払期限を示します。次のいずれかの値になります。
  - 0：支払が後払い（期間内の最後）の場合
  - 1：支払が先払い（期間内の初め）の場合

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

### 注意:

支払う金額（Pmtパラメータの金額）はマイナスの数値、受け取る総額はプラスの数値で表わされます。

## Oct()

10進法のパラメータの8進法の値を返します。

## 内部BASICシンタックス

Function Oct(dValue As Double) As String

## 用途

バージョン：3.00

|                        | 使用                                                                                  |
|------------------------|-------------------------------------------------------------------------------------|
| AssetCenter API        |                                                                                     |
| リンクまたはフィールドの設定スクリプト    |  |
| 「スクリプト」タイプのアクションの設定    |  |
| ウィザードスクリプト             |  |
| ウィザードの「FINISH.DO」スクリプト |  |

## パラメータ

- dValue：8進法の値を取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dSeed as Double
dSeed = Int((10*Rnd)-5)
RetVal = Oct(dSeed)
```

## ParseDate()

文字列フォーマットの日付を、Basic日付オブジェクトへ変換します。

## 内部BASICシンタックス

Function ParseDate(strDate As String, strFormat As String, strStep As String) As Date

## 用途

バージョン : 3.60

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strDate : 文字列フォーマットの日付
- strFormat : 文字列に含まれている日付のフォーマット。以下の値が使用可能です。
  - DD/MM/YY
  - DD/MM/YYYY
  - MM/DD/YY
  - MM/DD/YYYY
  - YYYY/MM/DD
- Date : クライアントコンピュータの日付パラメータに基づいた日付
- DateInter : 国際標準形式の日付
- strStep : このオプションパラメータには、文字列内で使用される日付の区切り文字が含まれます。許可される区切り文字は「/」と「-」です。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dDate as date  
dDate=ParseDate("2001/05/01", "YYYY/MM/DD")
```

## ParseDMYDate()

次の形式の日付から、日付オブジェクト（BASIC形式）を返します。

```
dd/mm/yyyy
```

## 内部BASICシンタックス

Function ParseDMYDate(strDate As String) As Date

### 用途

バージョン : 3.5

#### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

### パラメータ

- strDate : 文字列として保存されている日付

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## ParseMDYDate()

次の形式の日付から、日付オブジェクト（BASIC形式）を返します。

mm/dd/yyyy

## 内部BASICシンタックス

Function ParseMDYDate(strDate As String) As Date

### 用途

バージョン : 3.5

#### 使用

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strDate : 文字列として保存されている日付

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## ParseYMDDate()

yyyy/mm/dd形式の日付の文字列を、BASICの日付型変数に変換します。

## 内部BASICシンタックス

Function ParseYMDDate(strDate As String) As Date

## 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strDate : 文字列として保存されている日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## PMT()

定額の定期的な支払、および一定利率に基づいて計算した、年間支払金額を返します。

## 内部BASICシンタックス

Function PMT(dblRate As Double, iNper As Long, dbIPV As Double, dbIFV As Double, iType As Long) As Double

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dblRate：支払日ごとの利率。例えば、年利が6%のローンの毎月の支払日の利率は、次のようになります。

|                        |
|------------------------|
| 0.06/12=0.005 または 0.5% |
|------------------------|

- iNper：総支払回数
- dbIPV：実際に支払わなければならない金額（総額）
- dbIFV：支払が終了した時の差引残高または将来の支払用に確保する金額。一般に、特にローンを返済する場合は、このパラメータを「0」に設定します。実際、すべての支払を済ませるとローンの値はゼロになります。
- iType：支払期限を示します。次のいずれかの値になります。
  - 0：支払が後払い（期間内の最後）の場合
  - 1：支払が先払い（期間内の初め）の場合

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注



注意:

- RateおよびNperパラメータの計算には同じ単位の支払金額を使用する必要があります。
- 支払う金額 (Pmtパラメータの金額) はマイナスの数値、受け取る総額はプラスの数値で表わされます。

## PPMT()

定額の定期的な支払い、および一定利率に基づき、指定した支払日に返済する元金の金額を返します。

## 内部BASICシンタックス

Function PPMT(dblRate As Double, iPer As Long, iNper As Long, dblPV As Double, dblFV As Double, iType As Long) As Double

## 用途

バージョン : 3.00

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト

「スクリプト」タイプのアクションの設定

ウィザードスクリプト

ウィザードの「FINISH.DO」スクリプト

## パラメータ

- dblRate : 支払日ごとの利率。例えば、年利が6%のローンの毎月の支払日の利率は、次のようになります。

0.06/12=0.005 または 0.5%

- iPer : 計算する期間。1からNperの数値の間で指定します。
- iNper : 総支払回数
- dbIPV : 実際に支払わなければならない金額 (総額)
- dbIFV : 支払が終了した時の差引残高または将来の支払用に確保する金額。一般に、特にローンを返済する場合は、このパラメータを「0」に設定します。実際、すべての支払を済ませるとローンの値はゼロになります。
- iType : 支払期限を示します。次のいずれかの値になります。
  - 0 : 支払が後払い (期間内の最後) の場合
  - 1 : 支払が先払い (期間内の初め) の場合

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数 (必要に応じて`AmLastErrorMsg()`関数も) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

---

 注意:

- RateおよびNperパラメータの計算には同じ単位の支払金額を使用する必要があります。
  - 支払う金額 (Pmtパラメータの金額) はマイナスの数値、受け取る総額はプラスの数値で表わされます。
- 

## PV()

定額の定期的な支払い、および固定利率に基づいて計算した、実際の年間支払総額を返します。

## 内部BASICシンタックス

Function PV(dblRate As Double, iNper As Long, dblPmt As Double, dblFV As Double, iType As Long) As Double

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dblRate：支払日ごとの利率。例えば、年利が6%のローンの毎月の支払日の利率は、次のようになります。

|                          |
|--------------------------|
| $0.06/12=0.005$ または 0.5% |
|--------------------------|

- iNper：総支払回数
- dblPmt：1回の支払金額。支払金額には一般的に元金と利子が含まれます。
- dblFV：支払が終了した時の差引残高または将来の支払用に確保する金額。一般に、特にローンを返済する場合は、このパラメータを「0」に設定します。実際、すべての支払を済ませるとローンの値はゼロになります。
- iType：支払期限を示します。次のいずれかの値になります。
  - 0：支払が後払い（期間内の最後）の場合
  - 1：支払が先払い（期間内の初め）の場合

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。

- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注

### 注意:

- RateおよびNperパラメータの計算には同じ単位の支払金額を使用する必要があります。
- 支払う金額（Pmtパラメータの金額）はマイナスの数値、受け取る総額はプラスの数値で表わされます。

## Randomize()

乱数発生関数を初期化します。

## 内部BASICシンタックス

Function Randomize(IValue As Long)

## 用途

バージョン：3.00

|                        | 使用                                                                                  |
|------------------------|-------------------------------------------------------------------------------------|
| AssetCenter API        |                                                                                     |
| リンクまたはフィールドの設定スクリプト    |  |
| 「スクリプト」タイプアクションの設定     |  |
| ウィザードスクリプト             |  |
| ウィザードの「FINISH.DO」スクリプト |  |

## パラメータ

- IValue : 特定の新しい初期値を指定して乱数発生関数であるRnd関数を初期化するときに使います。このパラメータを省略すると、システムクロックからの値が初期値として使われます。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyNumber
Randomize
MyNumber= Int((10*Rnd)+1) : ' 1 - 10の乱数値を返します。
RetVal=MyNumber
```

## RATE()

年間支払金額のうちの1回分の支払金額の利率を返します。

## 内部BASICシンタックス

Function RATE(iNper As Long, dbIPmt As Double, dbIFV As Double, dbIPV As Double, iType As Long, dbIGuess As Double) As Double

## 用途

バージョン : 3.00

## 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- iNper : 総支払回数
- dbIPmt : 1回の支払金額。支払いには一般的に元金と利子が含まれます。
- dbIFV : 支払が終了した時の差引残高または将来の支払用に確保する金額。一般に、特にローンを返済する場合は、このパラメータを「0」に設定します。実際、すべての支払を済ませるとローンの値はゼロになります。
- dbIPV : 実際に支払わなければならない金額 (総額)
- iType : 支払期限を示します。次のいずれかの値になります。
  - 0 : 支払が後払い (期間内の最後) の場合
  - 1 : 支払が先払い (期間内の初め) の場合
- dbIGuess : 1回の支払の利率の推定値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数 (必要に応じて`AmLastErrorMsg()`関数も) を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注



注意:

- 支払う金額 (Pmtパラメータの値) はマイナスの数値、受け取る総額はプラスの数値になります。
- この関数は、Guessパラメータに指定した値から開始して、繰り返し計算を実行します。20回繰り返しても結果が出ない場合は、この関数は無効となります。

## RemoveRows()

strRowNamesパラメータに指定されている行をリストから削除します。

この関数は、[ ListBox ] コントロールタイプの値を処理する時に役立ちます。

このタイプのコントロールの値は、次の文字で区切られています。

- パイプ文字 (|) は、列を区切ります。
- コンマ (,) は、行を区切ります。
- 各行の終わりには、等号 (=) とその後に固有のIDが付いています。

## 内部BASICシンタックス

Function RemoveRows(strList As String, strRowNames As String) As String

## 用途

バージョン : 3.5

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strList : 処理する [ ListBox ] コントロールの値が含まれている文字列
- strRowNames : 削除する行のID。IDが複数ある場合は、カンマで区切ります。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=RemoveRows("a1|a2=a0,b1|b2=b0", "a0,c0") : "b1|b2=b0"を返します。
RetVal=MyStr
```

## Replace()

strDataパラメータの文字列に含まれるstrOldPatternパラメータの文字列をすべてstrNewPatternパラメータの文字列で置き換えます。bCaseSensitiveパラメータを使って、検索するstrOldPatternパラメータの文字列の大文字 / 小文字を区別できます。

## 内部BASICシンタックス

Function Replace(strData As String, strOldPattern As String, strNewPattern As String, bCaseSensitive As Long) As String

## 用途

バージョン : 3.5

## 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプアクションの設定     | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strData : 置換される文字列を含んでいる文字列
- strOldPattern : strDataパラメータに含まれている検索の対象となる文字列。
- strNewPattern : 検索した文字列を置換する文字列
- bCaseSensitive : このパラメータを使って、大文字と小文字を区別する (=1)、または区別しない (=0) かを指定します。

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=Replace("youmeyoumeyou", "you", "me",0) : "mememememe"を返します。
MyStr=Replace("youmeyoumeyou", "You", "me",1) : "youmeyoumeyou"を返します
。
MyStr=Replace("youmeYoumeyou", "You", "me",1) : "youmememeyou"を返します
。
RetVal=""
```

## Right()

文字列の右端から iNumber で指定した数の文字を返します。

### 内部BASICシンタックス

Function Right(strString As String, iNumber As Long) As String

### 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strString : 処理する文字列
- iNumber : 返す文字数

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim lWord, strMsg, rWord, iPos : ' Declare variables.
strMsg = "Left() Test."
iPos = InStr(1, strMsg, " ") : ' スペースを探す
lWord = Left(strMsg, iPos - 1) : ' 左側の単語を取得
rWord = Right(strMsg, Len(strMsg) - iPos) : ' 右側の単語を取得
strMsg=rWord+lWord : ' 2つの単語を交換
RetVal=strMsg
```

## RightPart()

strSepパラメータに指定されている区切り文字の右側の文字列を1つ取得します。  
右から左に向かって区切り文字を探します。  
bCaseSensitiveパラメータを使って、大文字と小文字を区別することもできます。

## 内部BASICシンタックス

```
Function RightPart(strFrom As String, strSep As String, bCaseSensitive As Long) As String
```

## 用途

バージョン : 3.5

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strFrom : 処理する文字列
- strSep : 処理する文字列の区切り文字

- `bCaseSensitive` : このパラメータを使って、大文字と小文字を区別する (=1)、または区別しない (=0) を指定します。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

文字列"This\_is\_a\_test"に、`LeftPart`、`LeftPartFromRight`、`RightPart`、および`RightPartFromLeft`関数を使った例を示します。

```
LeftPart("This_is_a_test","_",0)
```

"This"を返します。

```
LeftPartFromRight("This_is_a_test","_",0)
```

"This\_is\_a"を返します。

```
RightPart("This_is_a_test","_",0)
```

"test"を返します。

```
RightPartFromLeft("This_is_a_test","_",0)
```

"is\_a\_test"を返します。

## RightPartFromLeft()

`strSep`パラメータに指定されている区切り文字の右側の文字列を1つ取得します。左から右に向かって区切り文字を探します。  
`bCaseSensitive`パラメータを使って、大文字と小文字を区別することもできます。

## 内部BASICシンタックス

Function RightPartFromLeft(strFrom As String, strSep As String, bCaseSensitive As Long) As String

### 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strFrom : 処理する文字列
- strSep : 処理する文字列の区切り文字
- bCaseSensitive : このパラメータを使って、大文字と小文字を区別する (=1)、または区別しない (=0) かを指定します。

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

文字列"This\_is\_a\_test"に、LeftPart、LeftPartFromRight、RightPart、およびRightPartFromLeft関数を使った例を示します。

```
LeftPart("This_is_a_test","_",0)
```

"This"を返します。

```
LeftPartFromRight("This_is_a_test","_",0)
```

"This\_is\_a"を返します。

```
RightPart("This_is_a_test","_",0)
```

"test"を返します。

```
RightPartFromLeft("This_is_a_test","_",0)
```

"is\_a\_test"を返します。

## Rmdir()

既存のディレクトリを1つ削除します。

## 内部BASICシンタックス

Function Rmdir(strRmDirectory As String) As Long

## 用途

バージョン : 3.00

### 使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト 

「スクリプト」タイプのアクションの設定 

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

---

## パラメータ

- strRmDirectory : 削除するディレクトリのフルパス

## 戻りコード

- 0 : 成功
- 0以外 : エラーコード

## Rnd()

乱数を含んでいる値を返します。

## 内部BASICシンタックス

Function Rnd(dValue As Double) As Double

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dValue : 関数の実行モードを定義する場合に使うパラメータ
  - 0より小さい値 : 関数を実行するたびに同じ数値を発生します。
  - 0より大きい値 : 次に発生する乱数
  - 0 : 直前に発生した乱数
  - 省略 : 次に発生する乱数

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 注



注意:

この関数を呼び出す前に、パラメータなしで`Randomize`関数を使い、乱数発生関数を初期化する必要があります。

## 例

```
Dim MyNumber
Randomize
MyNumber= Int((10*Rnd)+1) : ' 1 - 10のいずれかの値を乱数として返します。
RetVal=MyNumber
```

## RTrim()

文字列の末尾に含まれるスペースをすべて取り除きます。

## 内部BASICシンタックス

Function RTrim(strString As String) As String

## 用途

バージョン : 3.00

使用

AssetCenter API

[リンクまたはフィールドの設定スクリプト](#) 

|                        | 使用 |
|------------------------|----|
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strString : 処理する文字列

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```

This example uses the LTrim and RTrim functions to strip leading ' and trailing spaces
, respectively, from a string variable.
' It uses the Trim function alone to strip both types of spaces.
' LCase and UCase are also shown in this example as well as the use
' of nested function calls

Dim strString as String
Dim strTrimString as String
strString = " <-Trim-> " : ' Initialize string.
strTrimString = LTrim(strString) : ' strTrimString = "<-Trim-> ".
strTrimString = LCase(RTrim(strString)) : ' strTrimString = " <-trim->".
strTrimString = LTrim(RTrim(strString)) : ' strTrimString = "<-Trim->".
' Using the Trim function alone achieves the same result.
strTrimString = UCase(Trim(strString)) : ' strTrimString = "<-TRIM->".
RetVal = "|" & strTrimString & "|"

```

## Second()

tmTimeパラメータの時刻の秒の部分の数値を返します。

### 内部BASICシンタックス

Function Second(tmTime As Date) As Long

### 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- tmTime : 処理する「時刻+日付」形式のパラメータ

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strSecond
strSecond=Second(Date())
RetVal=strSecond : '現在の時刻の秒の部分を返します。例えば、時刻が15:45:30の場合は、"30"を返します。
```

## SetSubList()

[ ListBox ] コントロールのサブリストの値を定義します。

## 内部BASICシンタックス

Function SetSubList(strValues As String, strRows As String, strRowFormat As String)  
As String

## 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strValues : 処理する [ ListBox ] コントロールの値が含まれている文字列
- strRows : strValuesパラメータの文字列に追加する値またはその文字列に含まれている文字を置換する値の一覧。各値をパイプ文字 (|) で区切ります。処理する行は、等号 (=) 記号の右側にあるIDで識別されます。不明な行は処理されません。
- strRowFormat : サブリストの書式化命令。各命令をパイプ文字 (|) で区切ります。このパラメータには、次の文字を使います。

- 1は、サブリストの最初の列の情報を示します。
- i-jは、列のグループを定義します。
- ハイフン (-) は、すべての列を処理することを示します。
- 不明な列の値は返しません。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "A2|A1=a0, B2|B1=b0", "2|1") : ' "A1|A2|a3=a0,B1|B2|b3=b0,c1|c2|c3=c0"を返します。
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "Z2=*,B2=b0", "2") : ' "a1|Z2|a3=a0,b1|B2|b3=b0,c1|Z2|c3=c0"を返します。
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "B5|B6|B7=b0,C5|C6,C7=c0", "5-7") : ' "a1|a2|a3=a0,b1|b2|b3|B5|B6|B7=b0,c1|c2|c3||C5|C6|C7=c0"を返します。
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "B1|B2|B3|B4=b0", "-") : ' "a1|a2|a3=a0,B1|B2|B3|B4=b0,c1|c2|c3=c0"を返します。
MyStr=SubList("A|B|C,D|E|F", "X=*", "2") : ' "A|X|C,D|X|F"を返します。
RetVal=""
```

## Sgn()

数値の記号を表わす値を返します。

## 内部BASICシンタックス

Function Sgn(dValue As Double) As Double

## 用途

バージョン：3.00

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプアクションの設定     | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- dValue：記号を取得する数値

## 戻りコード

次のいずれかの値を返します。

- 1：0より大きい数値を示します。
- 0：0を示します。
- -1：0より小さい数値を示します。

## 例

```
Dim dNumber as Double
dNumber=-256
RetVal=Sgn(dNumber)
```

## Shell()

実行可能プログラムを起動します。

## 内部BASICシンタックス

Function Shell(strExec As String) As Long

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strExec：起動する実行可能ファイルのフルパス

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyId/h MyId=Shell("C:/WinNT/notepad.exe")/h RetVal=""
```

## Sin()

数値のサインを返します。単位はラジアンです。

## 内部BASICシンタックス

Function Sin(dValue As Double) As Double

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dValue : サインを取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dCalc as Double
dCalc=Sin(150)
RetVal=dCalc
```

## Space()

iSpaceパラメータに指定されている数のスペース（空白文字）を挿入した文字列を作成します。

## 内部BASICシンタックス

Function Space(iSpace As Long) As String

### 用途

バージョン : 3.00

#### 使用

| AssetCenter API        |   |
|------------------------|---|
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

### パラメータ

- iSpace : 文字列に挿入するスペースの数

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 注

#### 注意:

この関数は、文字列を書式化したり、固定長の文字列から日付を削除する時に使います。

## 例

```
Dim MyString
' 10個のスペースを返します。
MyString = Space(10)
' 2つの文字列の間に10個のスペースを挿入します。
MyString = "Space" & Space(10) & "inserted"
RetVal=MyString
```

## Sqr()

数値の平方根を返します。

## 内部BASICシンタックス

Function Sqr(dValue As Double) As Double

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- dValue : 平方根を取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dCalc as Double
dCalc=Sqr(81)
RetVal=dCalc
```

## Str()

数値を文字列に変換します。

## 内部BASICシンタックス

Function Str(strValue As String) As String

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- strValue：文字列に変換する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dNumber as Double
dNumber=Cos(150)
RetVal=Str(dCalc)
```

## StrComp()

2つの文字列を比較します。

## 内部BASICシンタックス

Function StrComp(strString1 As String, strString2 As String, iOptionCompare As Long)  
As Long

## 用途

バージョン：3.00

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strString1 : 最初の文字列
- strString2 : 2つめの文字列
- iOptionCompare : 比較のタイプ。バイナリの比較には「0」、テキストの比較には「1」を設定します。

## 戻りコード

- -1 : strString1はstrString2よりも大きい。
- 0 : strString1はstrString2と等しい。
- 1 : strString1はstrString2よりも小さい。

## String()

strString文字をiCount回繰り返した文字列を返します。

## 内部BASICシンタックス

Function String(iCount As Long, strString As String) As String

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- iCount : strString文字を繰り返す回数

- strString : 繰り返す文字

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim iCount as Integer
Dim strTest as String
strTest="T"
iCount=5
RetVal=String(iCount,strTest)
```

## SubList()

[ ListBox ] コントロールの値の文字列に含まれている値一覧のサブリストを返します。

## 内部BASICシンタックス

Function SubList(strValues As String, strRows As String, strRowFormat As String) As String

## 用途

バージョン : 3.5

### 使用

---

AssetCenter API

[リンクまたはフィールドの設定スクリプト](#) 

---

使用

「スクリプト」タイプのアクションの設定



ウィザードスクリプト



ウィザードの「FINISH.DO」スクリプト



## パラメータ

- strValues : 処理する [ ListBox ] コントロールの値が含まれている文字列
- strRows : サプリストに含める行のID。各IDをカンマ (,) で区切って指定します。特定のワイルドカード文字を使用できます。
  - (\*) は、サブリスト内のすべての行を含むことを示します。
  - 不明なIDを指定した場合は、サブリストの空の値が返されます。
- strRowFormat : サプリストの書式化命令。各命令をパイプ文字 (|) で区切ります。このパラメータには、次の文字を使います。
  - 1は、取得するサブリストのリストの最初の列の情報を示します。
  - 0は、取得するサブリストのリストの行のIDを示します。
  - アスタリスク (\*) は、行のIDを除くすべての列の情報を示します。
  - 不明な列の値は返しません。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "a0,b0,a0", "3|2|3") : ' a3|a2|
a3,b3|b2|b3,a3|a2|a3"を返します。
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "*", "*|0") : 'Returns "a1|a2|a3
|a0,b1|b2|b3|b0,c1|c2|c3|c0"
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "*", "*=0") : ' "a1|a2|a3=a0,b
1|b2|b3=b0,c1|c2|c3=c0"を返します。
```

```

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "*", "999=0") : ' "=a0,=b0,=c0"を返します。
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "z0", "*=0") : ' ""を返します。
MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "*", "=1") : ' "=a1,=b1,=c1"を返します。
MyStr=SubList("A|B|C,D|E|F", "*", "2=0") : ' "B,E"を返します。
RetVal=""

```

## Tan()

数値のタンジェントを返します。単位はラジアンです。

## 内部BASICシンタックス

Function Tan(dValue As Double) As Double

## 用途

バージョン : 3.00

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- dValue : タンジェントを取得する数値

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim dCalc as Double
dCalc=Tan(150)
RetVal=dCalc
```

## Time()

現在の時刻を返します。

## 内部BASICシンタックス

Function Time() As Date

## 用途

バージョン : 3.00

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Timer()

午前0時0分から経過した秒数を返します。

### 内部BASICシンタックス

Function Timer() As Double

### 用途

バージョン : 3.00

#### 使用

| AssetCenter API        |   |
|------------------------|---|
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## TimeSerial()

iHour、iMinute、iMinuteパラメータの形式に従って時刻を返します。

### 内部BASICシンタックス

Function TimeSerial(iHour As Long, iMinute As Long, iSecond As Long) As Date

### 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- iHour：時間
- iMinute：分
- iSecond：秒

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、[AmLastError\(\)](#)関数（必要に応じて[AmLastErrorMsg\(\)](#)関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

各パラメータにはそれぞれ、日、月、または年を表す数式を使用できます。

```
TimeSerial(12-8, -10, 0)
```

例えば、上記の例は次の値を返します。

```
3:50:00
```

パラメータの値が予想される範囲（分ならば0-59、時間ならば0-24など）以外の値の場合は、次のレベルのパラメータに繰り上げられます。つまり、iMinuteに75を入力すると、1時間と15分に解釈されます。

次の例は、

```
TimeSerial (16, 50, 45)
```

次の値を返します。

```
16:50:45
```

## TimeValue()

「日付 + 時刻」の値の時刻の部分の部分を返します。

## 内部BASICシンタックス

Function TimeValue(tmTime As Date) As Date

## 用途

バージョン : 3.00

### 使用

AssetCenter API

リンクまたはフィールドの設定スクリプト 

「スクリプト」タイプのアクションの設定 

ウィザードスクリプト 

ウィザードの「FINISH.DO」スクリプト 

## パラメータ

- tmTime : 「日付 + 時刻」形式の日付

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

次の例は、

```
TimeValue ("1999/09/24 15:00:00")
```

次の値を返します。

```
15:00:00
```

## ToSmart()

ソース文字列の各語の始めを大文字にします。

## 内部BASICシンタックス

```
Function ToSmart(strString As String) As String
```

## 用途

バージョン : 3.5

使用

---

AssetCenter API

リンクまたはフィールドの設定スクリプト 

---

## 使用

|                        |   |
|------------------------|---|
| 「スクリプト」タイプアクションの設定     | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strString : 処理するソース文字列

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Trim()

先頭と末尾のスペースを削除した文字列を返します。

## 内部BASICシンタックス

Function Trim(strString As String) As String

## 用途

バージョン : 3.00

## 使用

|                     |   |
|---------------------|---|
| AssetCenter API     |   |
| リンクまたはフィールドの設定スクリプト | ✓ |
| 「スクリプト」タイプアクションの設定  | ✓ |
| ウィザードスクリプト          | ✓ |



## パラメータ

- strString : 処理する文字列

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
' This example uses the LTrim and RTrim functions to strip leading ' and trailing spaces
', respectively, from a string variable.
' It uses the Trim function alone to strip both types of spaces.
' LCase and UCase are also shown in this example as well as the use
' of nested function calls

Dim strString as String
Dim strTrimString as String
strString = " <-Trim-> " : ' Initialize string.
strTrimString = LTrim(strString) : ' strTrimString = "<-Trim-> ".
strTrimString = LCase(RTrim(strString)) : ' strTrimString = " <-trim->".
strTrimString = LTrim(RTrim(strString)) : ' strTrimString = "<-Trim->".
' Using the Trim function alone achieves the same result.
strTrimString = UCase(Trim(strString)) : ' strTrimString = "<-TRIM->".
RetVal= "|" & strTrimString & "|"
```

## UCase()

文字列に含まれるすべての小文字を大文字に変換して返します。

## 内部BASICシンタックス

Function UCase(strString As String) As String

### 用途

バージョン : 3.00

#### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

### パラメータ

- strString : 大文字に変換する文字列

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

### 例

```
' This example uses the LTrim and RTrim functions to strip leading ' and trailing spaces
', respectively, from a string variable.
' It uses the Trim function alone to strip both types of spaces.
' LCase and UCase are also shown in this example as well as the use
' of nested function calls

Dim strString as String
```

```
Dim strTrimString as String
strString = " <-Trim-> " : ' Initialize string.
strTrimString = LTrim(strString) : ' strTrimString = "<-Trim-> ".
strTrimString = LCase(RTrim(strString)) : ' strTrimString = " <-trim->".
strTrimString = LTrim(RTrim(strString)) : ' strTrimString = "<-Trim->".
' Using the Trim function alone achieves the same result.
strTrimString = UCase(Trim(strString)) : ' strTrimString = "<-TRIM->".
RetVal= "|" & strTrimString & "|"
```

## UnEscapeSeparators()

1つの文字列からすべてのエスケープ文字を削除します。

### 内部BASICシンタックス

Function UnEscapeSeparators(strSource As String, strEscChar As String) As String

### 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

### パラメータ

- strSource : 処理する文字列
- strEscChar : 削除するエスケープ文字

### 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=UnEscapeSeparators("you/me/you/", "/") : "you|me|you|"を返します。
RetVal=""
```

## Union()

区切り文字で区切られている2つの文字列を合体します。重複する文字列は削除されます。

## 内部BASICシンタックス

Function Union(strListOne As String, strListTwo As String, strSeparator As String, strEscChar As String) As String

## 用途

バージョン : 3.5

### 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strListOne : 最初の文字列
- strListTwo : 2番目の文字列
- strSeparator : 各文字列の区切り文字
- strEscChar : エスケープ文字。この文字が区切り文字の前に付くと、その区切り文字は無視されます。

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim MyStr
MyStr=Union("a1|a2,b1|b2", "a1|a3,b1|b2", ",", "/" ) : "a1|a2,b1|b2,a1|a3"を返します
。
MyStr=Union("a1|a2,b1|b2", "a1|a3,/b1|b2", ",", "/" ) : "a1|a2,b1|b2,a1|a3,/b1|b2"を返
します。
RetVal=""
```

## UTCToLocalDate()

UTCフォーマットの日付（タイムゾーンに関係しない）「日付 + 時刻」型の日付に変換します。

## 内部BASICシンタックス

```
Function UTCToLocalDate(tmUTC As Date) As Date
```

## 用途

バージョン : 3.5

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプアクションの設定     | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- tmUTC : UTCフォーマットの日付

## 戻りコード

- エラーの場合は、次の2つの処理方法を用意します。
- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
  - 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## Val()

数値を表す文字列を倍精度型に変換します。

## 内部BASICシンタックス

Function Val(strString As String) As Double

## 用途

バージョン : 3.00

## 使用

|                        |   |
|------------------------|---|
| AssetCenter API        |   |
| リンクまたはフィールドの設定スクリプト    | ✓ |
| 「スクリプト」タイプのアクションの設定    | ✓ |
| ウィザードスクリプト             | ✓ |
| ウィザードの「FINISH.DO」スクリプト | ✓ |

## パラメータ

- strString : 変換する文字列

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、`AmLastError()`関数（必要に応じて`AmLastErrorMsg()`関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。

## 例

```
Dim strYear
Dim dYear as Double
strYear=Year(Date())
dYear=Val(strYear)
RetVal=dYear :Returns the current year
```

## WeekDay()

tmDateパラメータの日付の曜日の部分を返します。

## 内部BASICシンタックス

Function WeekDay(tmDate As Date) As Long

## 用途

バージョン : 3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- tmDate : 処理する「時刻+日付」形式のパラメータ

## 戻りコード

「1」は日曜日、「2」は月曜日、...、「7」は土曜日のように、週の曜日に対応した数値を返します。

## 例

```
Dim strWeekDay
strWeekDay=WeekDay(Date())
RetVal=strWeekDay : '曜日を返します。
```

## Year()

tmDateパラメータの日付の年の部分を返します。

## 内部BASICシンタックス

Function Year(tmDate As Date) As Long

## 用途

バージョン：3.00

|                        | 使用 |
|------------------------|----|
| AssetCenter API        |    |
| リンクまたはフィールドの設定スクリプト    | ✓  |
| 「スクリプト」タイプのアクションの設定    | ✓  |
| ウィザードスクリプト             | ✓  |
| ウィザードの「FINISH.DO」スクリプト | ✓  |

## パラメータ

- tmDate：処理する「時刻+日付」形式のパラメータ

## 戻りコード

エラーの場合は、次の2つの処理方法を用意します。

- AssetCenter内で、この関数を含むスクリプトを中断し、ユーザの画面にエラーメッセージを表示します。
- 外部プログラムからこの関数を呼び出した場合は、AmLastError()関数（必要に応じてAmLastErrorMsg()関数も）を呼び出し、エラーが発生したかどうかを確認し、エラーが発生した場合はエラーメッセージを返すようにします。



# IV | 索引

---

パート



# 使用可能な関数 - 機能：全種

- Abs
- AmActionDde
- AmActionExec
- AmActionMail
- AmActionPrint
- AmActionPrintPreview
- AmActionPrintTo
- AmAddAllPOLinesToInv
- AmAddCatRefAndCompositionToPOOrder
- AmAddCatRefToPOOrder
- AmAddEstimLinesToPO
- AmAddEstimLineToPO
- AmAddPOLineToInv
- AmAddPOOrderLineToReceipt
- AmAddReceiptLineToInvoice
- AmAddReqLinesToEstim
- AmAddReqLinesToPO
- AmAddReqLineToEstim
- AmAddReqLineToPO
- AmAddRequestLineToPOOrder
- AmAddTemplateToPOOrder
- AmAddTemplateToRequest
- AmBusinessSecondsInDay
- AmCalcConsolidatedFeature
- AmCalcDepr
- AmCbkJReplayEvent
- AmCheckTraceDone
- AmCleanup
- AmClearLastError
- AmCloseAllChildren
- AmCloseConnection
- AmCommit
- AmComputeAllLicAndInstallCounts
- AmComputeLicAndInstallCounts
- AmConnectTrace
- AmConvertCurrency
- AmConvertDateBasicToUnix
- AmConvertDateIntlToUnix
- AmConvertDateStringToUnix
- AmConvertDateUnixToBasic
- AmConvertDateUnixToIntl
- AmConvertDateUnixToString
- AmConvertDoubleToString
- AmConvertMonetaryToString
- AmConvertStringToDouble
- AmConvertStringToMonetary
- AmCounter
- AmCreateAssetPort
- AmCreateAssetsAwaitingDelivery
- AmCreateCable

- AmCreateCableBundle
- AmCreateCableLink
- AmCreateDelivFromPO
- AmCreateDevice
- AmCreateDeviceLink
- AmCreateEstimFromReq
- AmCreateEstimsFromAllReqLines
- AmCreateInvFromPO
- AmCreateLink
- AmCreatePOFromEstim
- AmCreatePOFromReq
- AmCreatePOOrderFromRequest
- AmCreatePOOrdersFromRequest
- AmCreatePOsFromAllReqLines
- AmCreateProjectCable
- AmCreateProjectDevice
- AmCreateProjectTrace
- AmCreateReceiptFromPOOrder
- AmCreateRecord
- AmCreateRequestToInvoice
- AmCreateRequestToPOOrder
- AmCreateRequestToReceipt
- AmCreateReturnFromReceipt
- AmCreateTraceHist
- AmCryptPassword
- AmCurrentDate
- AmCurrentIsoLang
- AmCurrentLanguage
- AmCurrentServerDate
- AmDateAdd
- AmDateAddLogical
- AmDateDiff
- AmDbGetDate
- AmDbGetDouble
- AmDbGetList
- AmDbGetListEx
- AmDbGetLong
- AmDbGetPk
- AmDbGetString
- AmDbGetStringEx
- AmDeadLine
- AmDecrementLogLevel
- AmDefAssignee
- AmDefaultCurrency
- AmDefEscalationScheme
- AmDefGroup
- AmDeleteLink
- AmDeleteRecord
- AmDisconnectTrace
- AmDuplicateRecord
- AmEndOfNthBusinessDay
- AmEnumValList
- AmEvalScript
- AmExecTransition
- AmExecuteActionById
- AmExecuteActionByName
- AmExportDocument
- AmFindCable
- AmFindDevice
- AmFindRootLink
- AmFindTermDevice
- AmFindTermField
- AmGenSqlName
- AmGetCatRef
- AmGetCatRefFromCatProduct
- AmGetComputeString
- AmGetCurrentNTDomain
- AmGetCurrentNTUser
- AmGetFeat
- AmGetFeatCount
- AmGetField
- AmGetFieldCount
- AmGetFieldDateValue
- AmGetFieldDescription

- AmGetFieldDoubleValue
- AmGetFieldFormat
- AmGetFieldFormatFromName
- AmGetFieldFromName
- AmGetFieldLabel
- AmGetFieldLabelFromName
- AmGetFieldLongValue
- AmGetFieldName
- AmGetFieldRights
- AmGetFieldSize
- AmGetFieldSqlName
- AmGetFieldStrValue
- AmGetFieldType
- AmGetFieldUserType
- AmGetForeignKey
- AmGetIndex
- AmGetIndexCount
- AmGetIndexField
- AmGetIndexFieldCount
- AmGetIndexFlags
- AmGetIndexName
- AmGetLink
- AmGetLinkCardinality
- AmGetLinkCount
- AmGetLinkDstField
- AmGetLinkFeatureValue
- AmGetLinkFromName
- AmGetLinkType
- AmGetMainField
- AmGetMemoField
- AmGetNextAssetPin
- AmGetNextAssetPort
- AmGetNextCableBundle
- AmGetNextCablePair
- AmGetNTDomains
- AmGetNTMachinesInDomain
- AmGetNTUsersInDomain
- AmGetPOLinePrice
- AmGetPOLinePriceCur
- AmGetPOLineReference
- AmGetRecordFromMainId
- AmGetRecordHandle
- AmGetRecordId
- AmGetRelDstField
- AmGetRelSrcField
- AmGetRelTable
- AmGetReverseLink
- AmGetSelfFromMainId
- AmGetSourceTable
- AmGetTable
- AmGetTableCount
- AmGetTableDescription
- AmGetTableFromName
- AmGetTableLabel
- AmGetTableName
- AmGetTableRights
- AmGetTableSqlName
- AmGetTargetTable
- AmGetTrace
- AmGetTraceFromHist
- AmGetTypedLinkField
- AmGetVersion
- AmHasAdminPrivilege
- AmHasRelTable
- AmImportDocument
- AmIncrementLogLevel
- AmInsertRecord
- AmInstantiateReqLine
- AmInstantiateRequest
- AmIsConnected
- AmIsFieldForeignKey
- AmIsFieldIndexed
- AmIsFieldPrimaryKey
- AmIsLink

- AmIsTypedLink
- AmLastError
- AmLastErrorMsg
- AmListToString
- AmLog
- AmLoginId
- AmLoginName
- AmMapSubReqLineAgent
- AmMoveCable
- AmMoveDevice
- AmMsgBox
- AmOpenConnection
- AmOpenScreen
- AmPagePath
- AmProgress
- AmQueryCreate
- AmQueryExec
- AmQueryGet
- AmQueryNext
- AmQuerySetAddMainField
- AmQuerySetFullMemo
- AmQueryStartTable
- AmQueryStop
- AmReceiveAllPOLines
- AmReceivePOLine
- AmRefreshAllCaches
- AmRefreshLabel
- AmRefreshProperty
- AmRefreshTraceHist
- AmReleaseHandle
- AmRemoveCable
- AmRemoveDevice
- AmReturnAsset
- AmReturnContract
- AmReturnPortfolioItem
- AmReturnTraining
- AmReturnWorkOrder
- AmRevCryptPassword
- AmRgbColor
- AmRollback
- AmSetFieldDateValue
- AmSetFieldDoubleValue
- AmSetFieldLongValue
- AmSetFieldStrValue
- AmSetLinkFeatureValue
- AmSetProperty
- AmShowCableCrossConnect
- AmShowDeviceCrossConnect
- AmSqlTextConst
- AmStartTransaction
- AmStartup
- AmTableDesc
- AmTaxRate
- AmUpdateDetail
- AmUpdateRecord
- AmValueOf
- AmWizChain
- AmWorkTimeSpanBetween
- AppendOperand
- ApplyNewVals
- Asc
- Atn
- BasicToLocalDate
- BasicToLocalTime
- BasicToLocalTimeStamp
- Beep
- CDbI
- ChDir
- ChDrive
- Chr
- Clnt
- CLng
- Cos
- CountOccurrences

- CountValues
- CSng
- CStr
- CurDir
- CVar
- DaContext
- DaCopy
- DaDbDeleteList
- DaDbGetList
- DaDbSetList
- DaDelete
- DaDownload
- DaDumpContext
- DaExec
- DaExecAction
- DaExecuteActionByName
- DaFileATime
- DaFileCRC
- DaFileCTime
- DaFileLanguage
- DaFileMTime
- DaFileSize
- DaFileType
- DaFileVersion
- DaFind
- DaFindNext
- DaFirstEnv
- DaGetEnv
- DaGetFileInfo
- DaImpersonate
- DaMkDir
- DaMove
- DaNetIpFromName
- DaNetNBTName
- DaNetPing
- DaNetWakeOnLan
- DaNetWinAddressByName
- DaNextEnv
- DaNTFileCopyTo
- DaNTFileCreateDir
- DaNTFileDelete
- DaNTFileDeleteDir
- DaNTFileDirCopyTo
- DaNTFileDirDownload
- DaNTFileDirUpload
- DaNTFileDownload
- DaNTFileUpload
- DaNTRegistryLMAddStringValue
- DaNTRegistryLMCreateKey
- DaNTRegistryLMDeleteKey
- DaNTRegistryLMDeleteValue
- DaNTRegistryLMGetLongValue
- DaNTRegistryLMGetStringValue
- DaNTRegistryLMSetLongValue
- DaNTRegistryLMSetStringValue
- DaNTServiceInstall
- DaNTServiceStart
- DaNTServiceStatus
- DaNTServiceStop
- DaNTServiceUninstall
- DaNTWMIExecMethod
- DaNTWMIExecQuery
- DaNTWMIGetCurrentArrayValue
- DaNTWMIGetCurrentPropertyValue
- DaNTWMIGetInstanceCount
- DaNTWMIGetPropertyValue
- DaNTWMIGetTotalPropertiesValue
- DaNTWMINextItem
- DaNTWMIResetEnumeration
- DaRegCreateKey
- DaRegDeleteKey
- DaRegExec
- DaRegGetValue
- DaRegOutputValue

- DaRegSetValue
- DaRegStrValue
- DaRegVarValue
- DaRename
- DaReturnValue
- DaRmdir
- DaSendMail
- DaSetContext
- DaSetOption
- DaSetReturnValue
- Date
- DateSerial
- DateValue
- DaTrackingDelete
- DaTrackingGet
- DaTrackingSet
- DaTrackingTest
- DaUpload
- DaWait
- Day
- EscapeSeparators
- ExeDir
- Exp
- ExtractValue
- FileCopy
- FileDateTime
- FileExists
- FileLen
- Fix
- FormatDate
- FormatResString
- FV
- GetListItem
- Hex
- Hour
- InStr
- Int
- IPMT
- IsNumeric
- Kill
- LCase
- Left
- LeftPart
- LeftPartFromRight
- Len
- LocalToBasicDate
- LocalToBasicTime
- LocalToBasicTimeStamp
- LocalToUTCDate
- Log
- LTrim
- MakeInvertBool
- Mid
- Minute
- Mkdir
- Month
- Name
- Now
- NPER
- Oct
- ParseDate
- ParseDMYDate
- ParseMDYDate
- ParseYMDDate
- PMT
- PPMT
- PV
- Randomize
- RATE
- RemoveRows
- Replace
- Right
- RightPart
- RightPartFromLeft

- Rmdir
- Rnd
- RTrim
- Second
- SetSubList
- Sgn
- Shell
- Sin
- Space
- Sqr
- Str
- StrComp
- String
- SubList
- Tan
- Time
- Timer
- TimeSerial
- TimeValue
- ToSmart
- Trim
- UCase
- UnEscapeSeparators
- Union
- UTCToLocalDate
- Val
- WeekDay
- Year



# 使用可能な関数 - 機能：調達

- AmAddAllPOLinesToInv
- AmAddCatRefAndCompositionToPOOrder
- AmAddCatRefToPOOrder
- AmAddEstimLinesToPO
- AmAddEstimLineToPO
- AmAddPOLineToInv
- AmAddPOOrderLineToReceipt
- AmAddReceiptLineToInvoice
- AmAddReqLinesToEstim
- AmAddReqLinesToPO
- AmAddReqLineToEstim
- AmAddReqLineToPO
- AmAddRequestLineToPOOrder
- AmAddTemplateToPOOrder
- AmAddTemplateToRequest
- AmCreateAssetsAwaitingDelivery
- AmCreateDelivFromPO
- AmCreateEstimFromReq
- AmCreateEstimsFromAllReqLines
- AmCreateInvFromPO
- AmCreatePOFromEstim
- AmCreatePOFromReq
- AmCreatePOOrderFromRequest
- AmCreatePOOrdersFromRequest
- AmCreatePOsFromAllReqLines
- AmCreateReceiptFromPOOrder
- AmCreateRequestToInvoice
- AmCreateRequestToPOOrder
- AmCreateRequestToReceipt
- AmCreateReturnFromReceipt
- AmGetCatRef
- AmGetCatRefFromCatProduct
- AmGetPOLinePrice
- AmGetPOLinePriceCur
- AmGetPOLineReference
- AmInstantiateReqLine
- AmInstantiateRequest
- AmMapSubReqLineAgent
- AmReceiveAllPOLines
- AmReceivePOLine
- AmReturnAsset
- AmReturnContract
- AmReturnPortfolioItem
- AmReturnTraining
- AmReturnWorkOrder



# 使用可能な関数 - 機能：ケーブル

## 目次

- AmCheckTraceDone
- AmConnectTrace
- AmCreateAssetPort
- AmCreateCable
- AmCreateCableBundle
- AmCreateCableLink
- AmCreateDevice
- AmCreateDeviceLink
- AmCreateProjectCable
- AmCreateProjectDevice
- AmCreateProjectTrace
- AmCreateTraceHist
- AmDisconnectTrace
- AmFindCable
- AmFindDevice
- AmFindRootLink
- AmFindTermDevice
- AmFindTermField
- AmGetNextAssetPin
- AmGetNextAssetPort
- AmGetNextCableBundle
- AmGetNextCablePair
- AmGetTrace
- AmGetTraceFromHist
- AmMoveCable
- AmMoveDevice
- AmRefreshLabel
- AmRefreshTraceHist
- AmRemoveCable
- AmRemoveDevice
- AmShowCableCrossConnect
- AmShowDeviceCrossConnect



# 使用可能な関数 - 機能：アクション

## 目次

- AmActionDde
- AmActionExec
- AmActionMail
- AmActionPrint
- AmActionPrintPreview
- AmActionPrintTo
- AmExecuteActionById
- AmExecuteActionByName



# 使用可能な関数 - 機能：組み込み

## 目次

- Abs
- AppendOperand
- ApplyNewVals
- Asc
- Atn
- BasicToLocalDate
- BasicToLocalTime
- BasicToLocalTimeStamp
- Beep
- CDbl
- ChDir
- ChDrive
- Chr
- CInt
- CLng
- Cos
- CountOccurrences
- CountValues
- CSng
- CStr
- CurDir
- CVar
- Date
- DateSerial
- DateValue
- Day
- EscapeSeparators
- ExeDir
- Exp
- ExtractValue
- FileCopy
- FileDateTime
- FileExists
- FileLen
- Fix
- FormatDate
- FormatResString
- FV
- GetListItem
- Hex
- Hour
- InStr
- Int
- IPMT
- IsNumeric
- Kill
- LCase
- Left
- LeftPart
- LeftPartFromRight

- Len
- LocalToBasicDate
- LocalToBasicTime
- LocalToBasicTimeStamp
- LocalToUTCDate
- Log
- LTrim
- MakeInvertBool
- Mid
- Minute
- Mkdir
- Month
- Name
- Now
- NPER
- Oct
- ParseDate
- ParseDMYDate
- ParseMDYDate
- ParseYMDDate
- PMT
- PPMT
- PV
- Randomize
- RATE
- RemoveRows
- Replace
- Right
- RightPart
- RightPartFromLeft
- Rmdir
- Rnd
- RTrim
- Second
- SetSubList
- Sgn
- Shell
- Sin
- Space
- Sqr
- Str
- StrComp
- String
- SubList
- Tan
- Time
- Timer
- TimeSerial
- TimeValue
- ToSmart
- Trim
- UCase
- UnEscapeSeparators
- Union
- UTCToLocalDate
- Val
- WeekDay
- Year

# 使用可能な関数 - 機能：ウィザード

## 目次

- AmDecrementLogLevel
- AmExecTransition
- AmIncrementLogLevel
- AmLog
- AmPagePath
- AmProgress
- AmRefreshProperty
- AmSetProperty
- AmUpdateDetail
- AmValueOf
- AmWizChain



# 使用可能な関数 - 機能：認証

- Dalmpersonate



# 使用可能な関数 - 機能：リモート実行

## 目次

- DaExec



# 使用可能な関数 - 機能：ファイル管理

## 目次

- DaCopy
- DaDelete
- DaDownload
- DaFileATime
- DaFileCRC
- DaFileCTime
- DaFileLanguage
- DaFileMTime
- DaFileSize
- DaFileType
- DaFileVersion
- DaFind
- DaFindNext
- DaGetFileInfo
- DaMkDir
- DaMove
- DaRename
- DaRmDir
- DaUpload



# 使用可能な関数 - 機能：コン テキスト

## 目次

- DaContext
- DaDumpContext
- DaSetContext



# 使用可能な関数 - 機能：レジ ストリ

## 目次

- DaRegCreateKey
- DaRegDeleteKey
- DaRegExec
- DaRegGetValue
- DaRegOutputValue
- DaRegSetValue
- DaRegStrValue
- DaRegVarValue



# 使用可能な関数 - 機能：データベース

## 目次

- DaDbDeleteList
- DaDbGetList
- DaDbSetList
- DaExecAction
- DaExecuteActionByName



# 使用可能な関数 - 機能：トラッキング

## 目次

- DaTrackingDelete
- DaTrackingGet
- DaTrackingSet
- DaTrackingTest



## 使用可能な関数 - 機能 : WMI

- DaNTWMIExecMethod
- DaNTWMIExecQuery
- DaNTWMIGetCurrentArrayValue
- DaNTWMIGetCurrentPropertyValue
- DaNTWMIGetInstanceCount
- DaNTWMIGetPropertyValue
- DaNTWMIGetTotalPropertiesValue
- DaNTWMINextItem
- DaNTWMIResetEnumeration



# 使用可能な関数 - 機能： ネットワーク

## 目次

- DaNetIpFromName
- DaNetNBTName
- DaNetPing
- DaNetWakeOnLan
- DaNetWinAddressByName







October 10, 2002